I. CREANGA C. REISCHER D. SIMOVICI

introducere algebrică în informatică

* *

limbaje formale

EDITURA JUNIMEA

introducere algebrică în informatică

limbaje formale

dr. doc. I. CREANGĂ
profesor universitar

dr. C. REISCHER
profesor universitar

dr. D. SIMOVICI
lector universitar

introducere algebrică în informatică

* *

limbaje formale

EDITURA JUNIMEA IAŞI
1974

PREFAȚĂ

Acest volum continuă seria noastră de lucrări intitulată „Introducere algebrică în informatică". El este consacrat studiului unor obiecte matematice care abstractizează proprietățile limbajelor de programare ale calculatoarelor electronice și ale fenomenelor lingvistice naturale.

În literatura de specialitate, care tratează probleme analoage, dată fiind dificultatea de demonstrare a unor rezultate, se recurge frecvent sau la simpla enunțare a acestora, sau la justificări aproximative. Punctul nostru de vedere a constat în prezentarea unor raționamente complete în toate cazurile.

În cadrul acestei cărți sînt trecute în revistă proprietățile claselor de limbaje din ierarhia introdusă de Noam Chomsky, care ocupă actualmente locul central în teoria limbajelor formale. Am expus, pentru fiecare tip de limbaje, atît mijloacele finitare de generare a lor (gramatici), cît și dispozitivele adecvate de recunoaștere (automate).

O serie de aspecte importante, ca de pildă studiul familiilor abstracte de limbaje, utilizarea seriilor formale în studiul limbajelor, evaluarea complexității limbajelor, sisteme Lindenmayer, etc. vor face obiectul unui viitor volum.

Lucrarea se adresează matematicienilor, informaticienilor și lingviștilor și este concepută astfel, încît lectura ei să nu impună, pe cît posibil, utilizarea altor surse. Sperăm că ea va contribui la completarea literaturii românești de informatică.

Autorii

— Lista de notaţii principale —

$\|M\|$	– numărul cardinal al mulţimii M
$\varnothing$	– mulţimea vidă
$\mathcal{P}(M)$	– mulţimea părţilor lui M
$\mathcal{P}^*(M)$	– mulţimea părţilor nevide ale lui M
χ_M	– funcţia caracteristică a mulţimii M
$\varphi \upharpoonright I$	– restricţia lui φ la I
I^*	– semigrupul liber generat de I
p, q	– cuvinte
e	– cuvînt nul
$\tilde{p}$	– cuvîntul obţinut din p prin operaţia de oglindire
$l(p)$	– lungimea cuvîntului p
N	– mulţimea numerelor naturale
N^*	– mulţimea numerelor naturale fără zero
$j = \overline{1, n}$	– indicele j ia valorile de la 1 la n
$\mathrm{Eq}(S)$	– laticea echivalenţelor pe S
$\mathbf{s}$	– funcţia de succesiune
$\mathbf{o}^k$	– funcţia nulă
$\mathbf{i}_m^k$	– funcţia de proiecţie
$\mathbf{S}^{n+1}$	– operator de superpoziţie
$\mathbf{R}$	– operator de recursie primitivă
$\mathbf{M}$	– operator de minimizare
G, G_1	– gramatici
I_N	– alfabet neterminal
I_T	– alfabet terminal
F	– mulţimea regulilor de generare
x_0	– simbol iniţial
$p \underset{G}{\Rightarrow} q$	– cuvîntul p generează direct cuvîntul q în gramatica G
$p \overset{\bullet}{\underset{G}{\Rightarrow}} q$	– cuvîntul p generează cuvîntul q în gramatica G

$L(G)$	– limbajul generat de gramatica G
$\mathcal{L}_j$	– familia limbajelor de tip j, $j = \overline{0,3}$
$\mathcal{R}(I)$	– mulțimea limbajelor regulate din I
$\mathcal{P}_f(Z^*)$	– mulțimea limbajelor finite pe alfabetul Z
A	– automat
I	– alfabet de intrare
S	– mulțime de stări
O	– alfabet de ieșire
$\delta_1 \vdash \delta_2$	– configurația instantanee δ_1 precede imediat configurația instantanee δ_2
$\delta_1 \overset{\bullet}{\vdash} \delta_2$	– configurația instantanee δ_1 precede configurația instantanee δ_2
b	– simbolul blanc
#	– simbol de marcaj
APD	– automat pushdown
s_0	– stare inițială
$L(APD)$	– limbajul recunoscut de un automat pushdown
AL	– automat de tip Leticevski
$L(AL)$	– limbaj reprezentat într-un automat de tip Leticevski
ALM	– automat liniar mărginit
$L(ALM)$	– limbaj acceptat de un automat liniar mărginit
MT	– mașină Turing
MTN	– mașină Turing nestaționară
$MTND$	– mașină Turing nedeterministă
$MTBI$	– mașină Turing bilateral infinită
$MTMMB$	– mașină Turing cu mai multe benzi
$MT_1 * MT_2$	– compunerea mașinilor Turing MT_1 și MT_2
$L(MT)$	– limbajul acceptat de o mașină Turing
■	– sfîrșit de demonstrație

PARTEA I

ELEMENTE DE ALGEBRĂ

§1. 0. INTRODUCERE

În această primă parte a lucrării sînt expuse o serie de elemente de algebră strict necesare înțelegerii materialului prezentat. Includerea acestor paragrafe se datorează dorinței autorilor de a face, în măsura în care acest lucru a fost posibil, ca cititorul să nu fie obligat să facă apel la alte surse.

§1. 1. NOȚIUNI DE TEORIA MULȚIMILOR

Vom expune aici cîteva elemente de teoria mulțimilor cu care ocazie vom preciza notațiile folosite în această lucrare. Noțiunea de mulțime nu va fi definită fiind considerată aici ca o noțiune primară.

O mulțime este descrisă printr-o proprietate a elementelor sale.

Dacă m este un element din mulțimea M se notează acest fapt cu $m \in M$ și se citește „m aparține lui M"; în caz contrar vom pune $m \notin M$.

Mulțimea M_1 este *inclusă* în mulțimea M_2 dacă din $m \in M_1$ rezultă $m \in M_2$. M_1 va fi numit *parte* a lui M_2 sau *submulțime* a lui M_2. Se scrie $M_1 \subseteq M_2$.

Dacă M_1 este inclus în M_2, dar există $m_2 \in M_2$ și $m_2 \notin M_1$, atunci M_1 este *inclusă strict* în M_2 și se notează $M_1 \subset M_2$.

Mulțimea vidă, notată cu $\varnothing$ este mulțimea care nu conține nici un element. Ea este parte, prin convenție, a oricărei mulțimi.

Dacă $m \in M$ atunci, evident, $\{m\} \subseteq M$, unde $\{m\}$ este mulțimea al cărui unic element este m.

Mulțimile M_1 și M_2 sînt *egale*, $M_1 = M_2$, dacă $M_1 \subseteq M_2$ și $M_2 \subseteq M_1$.

Vom nota cu $\mathcal{P}(M)$ mulțimea tuturor părților mulțimii M. Avem $\varnothing \in \mathcal{P}(M)$ și $M \in \mathcal{P}(M)$; $M_1 \in \mathcal{P}(M)$ dacă și numai dacă $M_1 \subseteq M$.

Fie M_1, M_2 două mulțimi.

Reuniunea mulțimilor M_1, M_2 este mulțimea

$$M_1 \cup M_2 = \{m \mid m \in M_1 \text{ sau } m \in M_2\}.$$

Intersecția mulțimilor M_1, M_2 este mulțimea

$$M_1 \cap M_2 = \{m \mid m \in M_1 \text{ și } m \in M_2\}.$$

Diferența mulțimilor M_1, M_2 este mulțimea

$$M_1 \setminus M_2 = \{m \mid m \in M_1 \text{ și } m \notin M_2\}.$$

Diferența simetrică a mulțimilor M_1, M_2 este mulțimea

$$M_1 \oplus M_2 = (M_1 \setminus M_2) \cup (M_2 \setminus M_1).$$

Dacă M_1 este o parte a mulțimii M *complementara mulțimii* M_1 *față de mulțimea* M sau, pe scurt, *complementara mulțimii* M_1 este mulțimea

$$M_1^c = \{m \mid m \in M,\ m \notin M_1\}.$$

Produsul cartezian al mulțimilor M_1, M_2,..., M_n este mulțimea:

$$M_1 \times \ldots \times M_n = \{(m_1, \ldots, m_n) \mid m_j \in M_j,\ j = \overline{1, n}\}.$$

Dacă $M_1 = \ldots = M_n = M$ vom pune $M_1 \times \ldots \times M_n = M^n$.

Definiția I.1.1. O *relație n-ară* pe mulțimile $M_1, \ldots, M_n$ este o parte ρ a produsului cartezian $M_1 \times \ldots \times M_n$, $\rho \subseteq M_1 \times \ldots \times M_n$. În mulțimea relațiilor $\mathcal{P}(M_1 \times \ldots \times M_n)$ vom distinge relația vidă $\varnothing$ și relația totală $\omega = M_1 \times \ldots \times M_n$.

În cazul $M_1 = \ldots = M_n = M$ se obțin relațiile n-are pe mulțimea M. În cazul particular $n = 2$ se obțin *relațiile binare* numite pe scurt *relații*.

Definiția I.1.2. *O aplicație* f *de la mulțimea* M_1 *la mulțimea* M_2, $f: M_1 \to M_2$ este o relație $f \subseteq M_1 \times M_2$ cu proprietatea:

$$(m_1, m_2'), (m_1, m_2'') \in f \text{ atunci } m_2' = m_2''.$$

Vom scrie de obicei în loc de $(m_1, m_2) \in f$, $m_2 = f(m_1)$.

Dacă din $f(m_1') = f(m_1'')$ rezultă $m_1' = m_1''$, f este o *aplicație injectivă* sau, pe scurt, o *injecție* sau încă o *corespondență biunivocă*.

Dacă $\forall m_2 \in M_2$ există $m_1 \in M_1$, cu $f(m_1) = m_2$, f este o *aplicație surjectivă*, sau, pe scurt, o *surjecție*.

Aplicația f este *bijectivă*, sau pe scurt o *bijecție* dacă ea este injectivă și surjectivă.

Definiția I.1.3. *Proiecția a* j*-a a produsului cartezian* este aplicația $\mathrm{pr}_j: M_1 \times \ldots \times M_n \to M_j$ dată de

$$\mathrm{pr}_j((m_1, \ldots, m_n)) = m_j.$$

Definiția I.1.4. *O operație n-ară* pe mulțimea M este o aplicație $f: M^n \to M$; n este *aritatea* operației f.

Dacă $n = 2$ obținem *operațiile binare*. Rezultatul aplicării operației f asupra lui m_1 și m_2, $f(m_1, m_2)$ se mai notează cu $m_1 f m_2$.

Ne vom ocupa în continuare de operații cu relații.

Definiția I.1.5. *Produsul a două relații* $\rho \subseteq M_1 \times M_2$, $\tau \subseteq M_2 \times M_3$ este relația $\rho\tau \subseteq M_1 \times M_3$ dată de:

$$\rho\tau = \{(m_1, m_3) \mid m_1 \in M_1,\ m_3 \in M_3,\ \exists m_2 \in M_2,\ (m_1, m_2) \in \rho \text{ și } (m_2, m_3) \in \tau\}.$$

Observația I.1.1. Produsul de relații este asociativ, adică dacă $\rho \subseteq M_1 \times M_2$, $\sigma \subseteq M_2 \times M_3$ și $\tau \subseteq M_3 \times M_4$ atunci

$$\rho(\sigma\tau) = (\rho\sigma)\tau.$$

Definiția I.1.6. *Relația diagonală* pe mulțimea M este relația

$$\iota = \{(m, m) \mid m \in M\}.$$

Se constată imediat că, pentru orice relație $\rho \subseteq M \times M$ avem

$$\rho\iota = \iota\rho = \rho.$$

Puterea a n-a a relației $\rho \subseteq M \times M$ se definește în mod inductiv punînd $\rho^0 = \iota$, $\rho^1 = \rho$ și $\rho^m = \rho^{m-1}\rho$.

Definiția I.1.7. *Relația inversă* a relației $\rho \subseteq M_1 \times M_2$ este relația $\rho^{-1} \subseteq M_2 \times M_1$ dată de:

$$\rho^{-1} = \{(m_2, m_1) \mid m_2 \in M_2,\ m_1 \in M_1,\ (m_1, m_2) \in \rho\}.$$

Definiția I.1.8. Fie $\rho \subseteq M \times M$. *Relația* ρ este:

reflexivă, dacă $\iota \subseteq \rho$,
simetrică, dacă $\rho^{-1} = \rho$,
tranzitivă, dacă $\rho^2 \subseteq \rho$,
antisimetrică, dacă $\rho \cap \rho^{-1} \subseteq \iota$.

Dacă ρ este reflexivă, simetrică și tranzitivă ea este o *relație de echivalență*. Vom nota cu $\mathrm{Eq}(M)$ mulțimea relațiilor de echivalență pe M.

Dacă ρ este reflexivă și tranzitivă ea este o *relație de preordine*.

Dacă ρ este o relație de preordine antisimetrică ea este o *relație de ordine*. Cuplul (M, ρ) se numește *mulțime parțial ordonată*.

Fie $(M, \leqslant)$ o mulțime parțial ordonată și $M_1 \subseteq M$. Dacă $(m', m'') \in \leqslant$ vom mai nota acest lucru cu $m' \leqslant m''$. Mulțimea majoranților mulțimii M_1 este mulțimea

$$M_1^{\uparrow} = \{m \mid \forall m_1 \in M_1,\ m_1 \leqslant m\}.$$

Mulțimea minoranților mulțimii M este mulțimea

$$M_1^{\downarrow} = \{m \mid \forall m_1 \in M_1,\ m \leqslant m_1\}.$$

Mulțimile $M_1 \cap M_1^s$ și $M_1 \cap M_1^i$ conțin cel mult un element.

Să presupunem că, de exemplu mulțimea $M_1 \cap M_1^s$ ar conține cel puțin două elemente fie ele m', m''. Atunci, cum $m' \in M_1$ și $m'' \in M_1^s$ avem $m' \leqslant m''$. Apoi, din $m'' \in M_1$ și $m' \in M_1^s$ urmează $m'' \leqslant m'$, de unde, din antisimetria relației „$\leqslant$" urmează $m' = m''$.

Dacă $M_1 \cap M_1^s \neq \emptyset$ ($M_1 \cap M_1^i \neq \emptyset$) unicul element al mulțimii este *cel mai mare* (*cel mai mic*) *element al mulțimii* M_1. Cel mai mare element al unei mulțimi se mai numește *ultim element*, iar cel mai mic element se numește și *prim element* al mulțimii.

Dacă $M_1^s \cap (M_1^s)^s \neq \emptyset$ atunci unicul element al acestei mulțimi este *supremumul mulțimii* M_1; îl vom nota cu $\sup M_1$.

Dacă $M_1^i \cap (M_1^i)^i \neq \emptyset$ unicul element al acestei mulțimi este *infimumul* mulțimii M_1, vom nota acest element cu $\inf M_1$.

Definiția I.1.9. O *partiție* a mulțimii M este o familie de mulțimi $\{M_j \mid j \in J\}$ având următoarele proprietăți:

1) $$M_j \neq \emptyset, \quad \forall j \in J.$$

2) $$j, k \in J, j \neq k \text{ implică } M_j \cap M_k = \emptyset,$$

3) $$\cup \{M_j \mid j \in J\} = M.$$

Vom nota cu $\Pi(M)$ mulțimea partițiilor mulțimii M.

Teorema I.1.1. *Există o corespondență biunivocă între partițiile unei mulțimi și relațiile de echivalență pe acea mulțime.*

Demonstrație. Fie $\rho \in \mathrm{Eq}(M)$. Să considerăm familia de mulțimi $\pi(\rho) = \{\rho(m) \mid m \in M\}$ unde $\rho(m) = \{m' \mid m' \in M, (m, m') \in \rho\}$. Familia $\{\rho(m) \mid m \in M\}$ formează o partiție a mulțimii M. Într-adevăr $\forall m \in M$ avem $\rho(m) \neq \emptyset$, deoarece $m \in \rho(m)$. Apoi, dacă $(m_1, m_2) \notin \rho$ avem $\rho(m_1) \cap \rho(m_2) = \emptyset$, deoarece, în caz contrar ar exista $m \in M$ astfel ca $(m_1, m) \in \rho$, $(m_2, m) \in \rho$ de unde, în baza proprietăților de simetrie și reflexivitate am obține $(m_1, m_2) \in \rho$.

Invers, fie $\pi = \{M_j \mid j \in J\}$ o partiție a mulțimii M. Să considerăm relația binară $\rho(\pi) = \{(m_1, m_2) \mid \exists j \in J, m_1, m_2 \in M_j\}$, care este o echivalență așa cum se constată imediat.

Ca atare aplicația $k : \mathrm{Eq}(M) \to \Pi(M)$ dată de $k(\rho) = \pi(\rho)$ este o aplicație surjectivă. Mai mult, ea este și injectivă deoarece relația de echivalență atașată unei partiții este unică și astfel k este o corespondență biunivocă.

Fie M o mulțime și ρ o relație de echivalență pe M.

Definiția I.1.10. *Mulțimea cît* a mulțimii M față de relația de echivalență ρ este mulțimea:

$$M/\rho = \{\rho(m) \mid m \in M\}.$$

Clasa de echivalență a elementului $m \in M$ este mulțimea $\rho(m)$ și orice element $m_1 \in \rho(m)$ se numește *reprezentant* al clasei $\rho(m)$.

Rangul relației de echivalență ρ este $|M/\rho|$.

Definiția I.1.11. *Echivalența nucleară a aplicației* $f: M_1 \to M_2$ este relația $\ker f \subseteq M_1 \times M_1$ dată de

$$\ker f = \{(m_1, m_2) \mid f(m_1) = f(m_2)\}.$$

Teorema I.1.2. *Echivalența nucleară a aplicației* f *este o relație de echivalență. Orice aplicație* $f: M_1 \to M_2$ *se scrie* $f = hgk$ *astfel încît diagrama*:

$$\begin{array}{ccc} M_1 & \xrightarrow{\quad f \quad} & M_2 \\ {\scriptstyle k}\downarrow & & \uparrow{\scriptstyle h} \\ M_1/\ker f & \xrightarrow[\quad g \quad]{} & f(M_1) \end{array}$$

să fie comutativă unde k *este o aplicație surjectivă,* $k(m) = \rho(m)$, g *este o aplicație bijectivă,* $g(\rho(m)) = f(m)$, *iar* h *este aplicația injectivă de incluziune* $h(f(m)) = f(m)$, unde $\rho = \ker f$.

Demonstrație. Este clar că relația $\ker f$ este reflexivă, simetrică și tranzitivă. Apoi $h(g(k(m))) = h(g(\rho(m))) = h(f(m)) = f(m)$, $\forall m \in M_1$ adică diagrama este comutativă.

Remarcăm că g este bine definită[1] deoarece, dacă $\rho(m_1) = \rho(m)$ avem $(m_1, m) \in \ker f$, adică $f(m_1) = f(m)$; g este surjectivă deoarece fiecărei valori $f(m)$ îi corespunde o clasă de echivalență $\rho(m)$ și este injectivă deoarece din $f(m_1) = f(m)$ urmează $(m_1, m) \in \ker f$ sau $\rho(m_1) = \rho(m)$. ■

Exemplul I.1.1. Fie $M_1 \in M$. *Funcția caracteristică* a mulțimii M_1 este aplicația $\chi_{M_1}: M \to \{0,1\}$ dată de

$$\chi_{M_1}(m) = \begin{cases} 0, & \text{dacă } m \notin M_1 \\ 1, & \text{dacă } m \in M_1. \end{cases}$$

Remarcăm că mulțimea cît $M/\ker \chi_{M_1}$ constă din două clase M_1 și $M \setminus M_1$. Astfel rangul relației $\ker \chi_{M_1}$ este egal cu 2.

Fie M o mulțime și $\mathcal{P}(M)$ mulțimea părților sale.

Definiția I.1.12. Un *sistem de închidere* este o familie de părți $\mathbf{C} \in \mathcal{P}(M)$ cu proprietățile:

1) $\forall D,\ D \subseteq \mathbf{C}, \quad \bigcap \{M_1 \mid M_1 \in D\} \in \mathbf{C}$,

2) $M \in \mathbf{C}$.

Exemplul I.1.2. Mulțimea relațiilor de echivalență pe o mulțime M formează un sistem de închidere deoarece intersecția unei familii oarecare de relații de echivalență este o relație de echivalență. De asemenea mulțimile relațiilor reflexive sau tranzitive sau reflexive și tranzitive formează sisteme de închidere.

[1] Definirea lui g nu depinde de reprezentantul ales al clasei $\rho(m)$.

Definiția I.1.13. Un *operator de închidere* pe o mulțime M este o aplicație $\mathbf{c} : \mathcal{P}(M) \to \mathcal{P}(M)$ cu proprietățile :

1) $M_1 \subseteq M_2$ implică $\mathbf{c}(M_1) \subseteq \mathbf{c}(M_2)$, $\forall M_1, M_2 \in \mathcal{P}(M)$.

2) $M_1 \subseteq \mathbf{c}(M_1)$, $\forall M_1 \in \mathcal{P}(M)$,

3) $\mathbf{c}(\mathbf{c}(M_1)) = \mathbf{c}(M_1)$, $\forall M_1 \in \mathcal{P}(M)$.

Legătura dintre sistemele de închidere și operatorii de închidere este precizată în

Teorema I.1.3. *Orice sistem de închidere* $\mathbf{C}$ *definește un operator de închidere prin*

$$\mathbf{c}_{\mathbf{C}}(L) = \cap \{M_1 \mid M_1 \in \mathbf{C},\ L \subseteq M_1\}. \tag{I.1.1}$$

Invers, orice operator de închidere $\mathbf{c}$ *pe* M *definește un sistem de închidere* $\mathbf{C}_{\mathbf{c}} = \{M_1 \mid M_1 \in M,\ \mathbf{c}(M_1) = M_1\}$. *Aplicația* $f : SI \to OI$, *unde* SI *este mulțimea sistemelor de închidere ale mulțimii* M. OI *mulțimea operatorilor de închidere și* $f(\mathbf{C}) = \mathbf{c}_{\mathbf{C}}$ *este o bijecție.*

Demonstrație. În baza egalității (I.1.1) operatorul $\mathbf{c}_{\mathbf{C}}$ satisface primele două condiții din definiția I.1.13. Apoi $\mathbf{c}_{\mathbf{C}}(M_1) = M_1$ dacă și numai dacă $M_1 \in \mathbf{C}$. Într-adevăr dacă $M_1 \in \mathbf{C}$ cum $M_1 \subseteq M_1$ avem $\mathbf{c}_{\mathbf{C}}(M_1) \subseteq M_1$ și proprietatea 2) implică $\mathbf{c}_{\mathbf{C}}(M_1) = M_1$, iar dacă $\mathbf{c}_{\mathbf{C}}(M_1) = M_1$ avem $M_1 \in \mathbf{C}$ deoarece $\mathbf{C}$ este sistem de închidere și $\mathbf{c}_{\mathbf{C}}(M_1) \in \mathbf{C}$.

De aceea, cum $\mathbf{c}_{\mathbf{C}}(L) \in \mathbf{C}$, $\forall L \in \mathcal{P}(M)$ obținem $\mathbf{c}_{\mathbf{C}}(\mathbf{c}_{\mathbf{C}}(L)) = \mathbf{c}_{\mathbf{C}}(L)$, adică $\mathbf{c}_{\mathbf{C}}$ este un operator de închidere.

Invers, fie $\mathbf{c}$ un operator de închidere și fie $\mathbf{D} \subseteq \mathbf{C}_{\mathbf{c}}$ o familie de mulțimi. Să arătăm că

$$\mathbf{c}(\cap \{M_1 \mid M_1 \in \mathbf{D}\}) = \cap \{M_1 \mid M_1 \in \mathbf{D}\}.$$

Va fi suficient să verificăm că $\mathbf{c}(\cap \{M_1 \mid M_1 \in \mathbf{D}\}) \subseteq \cap \{M_1 \mid M_1 \in \mathbf{D}\}$. Avem $\mathbf{c}(\cap \{M_1 \mid M_1 \in \mathbf{D}\}) \subseteq \mathbf{c}(M_1) = M_1$, $\forall M_1 \in \mathbf{D}$. De aici obținem incluziunea dorită.

În fine $\mathbf{c}(M) = M$ deoarece $M \subseteq \mathbf{c}(M) \subseteq M$.

Aplicația $f : SI \to OI$ este evident surjectivă în baza celor de mai sus. Să arătăm că ea este injectivă adică $\mathbf{c}_{\mathbf{C}_1} = \mathbf{c}_{\mathbf{C}_2}$ implică $\mathbf{C}_1 = \mathbf{C}_2$. Va fi suficient să arătăm că $\mathbf{C}_{\mathbf{c}_{\mathbf{C}}} = \mathbf{C}$.

Într-adevăr $\mathbf{C}_{\mathbf{c}_{\mathbf{C}}} = \{M_1 \mid M_1 \subseteq M,\ \mathbf{c}_{\mathbf{C}}(M_1) = M_1\}$. În baza unei observații făcute în cursul demonstrației avem $\mathbf{C}_{\mathbf{c}_{\mathbf{C}}} \subseteq \mathbf{C}$. Dacă $M_1 \in \mathbf{C}$ avem $\mathbf{c}_{\mathbf{C}}(M_1) = M_1$ adică $M_1 \in \mathbf{C}_{\mathbf{c}_{\mathbf{C}}}$. ■

Exemplul I.1.3. Fie $\mathbf{C}_{RT}$ sistemul de închidere al relațiilor reflexive și tranzitive pe mulțimea M și $\mathbf{c}$ operatorul de închidere corespunzător. Atunci

$$\mathbf{c}(\alpha) = \bigcup_{j=0}^{\infty} \alpha^j,$$

unde $\alpha \subseteq M \times M$. Avem $\bigcup_{j=0}^{\infty} \alpha^j \subseteq \alpha^0 = \iota$ și $(\bigcup_{j=0}^{\infty} \alpha^j)(\bigcup_{j=0}^{\infty} \alpha^j) \subseteq \bigcup_{j=0}^{\infty} \alpha^j$ deci $\bigcup_{j=0}^{\infty} \alpha^j \in \mathbf{C}_{RT}$ și $\alpha \subseteq \bigcup_{j=0}^{\infty} \alpha^j$. De aceea $\mathbf{e}(\alpha) \subseteq \bigcup_{j=0}^{\infty} \alpha^j$. Fie acum $(m', m'') \in \alpha^j$. Există $\{m_k \mid k \in \overline{1, j-1}\} \subseteq M$ astfel încît $\{(m', m_1), (m_1, m_2), \ldots (m_{j-1}, m'')\} \subseteq \alpha \subseteq \mathbf{e}(\alpha)$. Cum $\mathbf{e}(\alpha)$ este o relație tranzitivă obținem $(m', m'') \in \mathbf{e}(\alpha)$ și dacă $\alpha^j \subseteq \mathbf{e}(\alpha)$, $\forall j \in N$ de unde $\bigcup_{j=0}^{\infty} \alpha^j \subseteq \mathbf{e}(\alpha)$.

Definiția I.1.14. *Închiderea reflexivă* și *tranzitivă* a relației α este relația

$$\mathbf{e}(\alpha) = \bigcup_{j=0}^{\infty} \alpha^j.$$

Noțiunea de graf de care ne vom ocupa în continuare permite o reprezentare „geometrică" pentru relații.

Definiția I.1.15. Un *graf orientat* este o pereche $\mathfrak{G} = (V, \Gamma)$ unde V este o mulțime, iar Γ o aplicație, $\Gamma : V \to \mathcal{P}(V)$. Mulțimea V se *numește mulțimea vîrfurilor grafului* $\mathfrak{G}$ sau *mulțimea nodurilor grafului* $\mathfrak{G}$ iar dacă $v_2 \in \Gamma(v_1)$ perechea ordonată (v_1, v_2) este un *arc în graf*.

Dacă $|V| < \infty$ graful $\mathfrak{G}$ este *finit*; în caz contrar $\mathfrak{G}$ este un *graf infinit*.

Grafurile se reprezintă atașînd fiecărui vîrf un punct din plan și fiecărui arc (v_1, v_2) o săgeată care unește punctele v_1 și v_2 îndreptată de la v_1 la v_2.

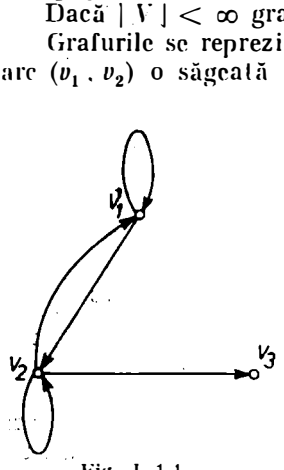

Fig. I. 1.1.

Exemplul I.1.4. Fie $\mathfrak{G} = (\{v_1, v_2, v_3\}, \Gamma)$ unde Γ este dat de $\Gamma(v_1) = \{v_1, v_2\}$, $\Gamma(v_2) = \{v_1, v_2, v_3\}$, $\Gamma(v_3) = \varnothing$. Graful $\mathfrak{G}$ este reprezentat în figura I. 1.1.

Definiția I.1.16. Un *subgraf al grafului* (V, Γ) este un graf (V_1, Γ_1) cu $V_1 \subseteq V$ și $\Gamma_1(v) = \Gamma(v) \cap V_1$.

Un *graf parțial* al grafului (V, Γ) este un graf de forma (V, Δ) cu $\Delta(v) \subseteq \Gamma(v)$, $\forall v \in V$.

Un *subgraf parțial* al grafului (V, Γ) este un graf (V_1, Δ_1) cu $V_1 \subseteq V$ și $\Delta_1(v) \subseteq \Gamma(v) \cap V_1$.

Exemplul I.1.5. Fie $\mathfrak{G} = (\{v_1, v_2, v_3\}, \Gamma)$ graful din exemplul I.1.4. Graful $(\{v_1, v_2\}, \Gamma_1)$ cu $\Gamma_1(v_1) = \Gamma_1(v_2) = \{v_1, v_2\}$ este un subgraf al grafului $\mathfrak{G}$.

Graful $(\{v_1, v_2, v_3\}, \Delta)$ cu $\Delta(v_1) = \{v_1\}$, $\Delta(v_2) = \{v_1, v_3\}$, $\Delta(v_3) = \varnothing$ este un graf parțial al grafului $\mathfrak{G}$.

Graful $(\{v_2, v_3\}, \Delta_1)$ cu $\Delta_1(v_2) = \{v_2\}$, $\Delta_1(v_3) = \varnothing$ este un subgraf parțial al grafului $\mathfrak{G}$.

Definiția I.1.17. *Mulțimea arcelor* grafului $\mathfrak{G} = (V, \Gamma)$ este mulțimea $U \subseteq V \times V$ dată de $U = \{(v_1, v_2) \mid v_2 \in \Gamma(v_1)\}$.

În arcul (v_1, v_2) vîrful v_1 este *extremitatea inițială*, iar vîrful v_2 este *extremitatea finală* a arcului (v_1, v_2).

Definiția I.1.18. Un *drum* de la vîrful v' la vîrful v'' în graful $\mathfrak{G}$ este o mulțime de arce $(v_1, v_2), (v_2, v_3), \ldots, (v_{n-1}, v_n)$ cu $v' = v_1$ și $v_n = v''$. Un *cir-*

cuit este un drum pentru care $v_1 = v_n$. Numărul $n-1$ este *lungimea drumului*. Un circuit de lungime 1 se numește *buclă*.

Definiția I.1.19. Un *graf conex* este un graf $\mathfrak{G} = (V, \Gamma)$ pentru care oricare ar fi vîrfurile v', $v'' \in V$ cu $v' \neq v''$ există un drum de la v' la v'' sau de la v'' la v'. Graful este *tare conex* dacă oricare ar fi vîrfurile v', $v'' \in V$ cu $v' \neq v''$ există un drum de la v' la v''.

Definiția I.1.20. *Componenta conexă* a vîrfului v este mulțimea de vîrfuri $C_v = \{v\} \cup \{v' \mid$ există un drum de la v' la v sau de la v' la $v\}$.

Observația I.1.2. Să considerăm relația $\rho \subseteq V \times V$ dată de

$$\rho = \{(v_1, v_2) \mid v_1 \in C_{v_2}\}.$$

Se constată imediat că ρ este o relație de echivalență și că $V/\rho = \{C_v \mid v \in V\}$. Astfel componentele conexe ale vîrfurilor formează o partiție a lui V.

Pentru un graf conex avem $|V/\rho| = 1$.

Definiția I.1.21. Un *arbore* este un graf conex $\mathfrak{G} = (V, \Gamma)$, fără circuite, avînd $|V| \geqslant 2$ și care satisface la următoarele două condiții:

i) există un vîrf v_0 care nu este extremitate finală a vreunui arc; v_0 se numește *rădăcina arborelui*;

ii) orice vîrf v, $v \in V \setminus \{v_0\}$ este extremitate finală a unui singur arc.

Nodurile v pentru care $\Gamma(v) = \varnothing$ se numesc *noduri finale*.

Exemplul I.1.6. Fie graful $\mathfrak{G} = (\{v_0, v_1, v_2, v_3, v_4, v_5\}, \Gamma)$ cu $\Gamma(v_0) = \{v_1, v_4\}$, $\Gamma(v_1) = \{v_2, v_3\}$, $\Gamma(v_2) = \Gamma(v_3) = \varnothing$, $\Gamma(v_4) = \{v_5\}$, $\Gamma(v_5) = \varnothing$, reprezentat în fig. I.1.2.

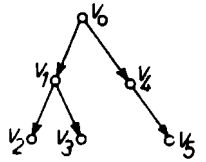

Fig. I.1.2.

Graful $\mathfrak{G}$ este un arbore avînd rădăcina v_0 și vîrfurile finale v_2, v_3 și v_5.

Definiția I.1.22. Vîrful v_2 este *descendent* direct al vîrfului v_1 în arborele $\mathfrak{G} = (V, \Gamma)$ dacă $(v_1, v_2) \in \Gamma \cup \{v\}$; cu alte cuvinte v_2 este un descendent direct al lui v_1 dacă există arcul (v_1, v_2) în graf, sau $v_2 = v_1$; în primul caz v_2 este un *descendent propriu*, iar în al doilea caz un *descendent impropriu* al vîrfului v_1.

Vîrful v'' este *descendent* al vîrfului v' în arborele $\mathfrak{G} = (V, \Gamma)$ dacă există un șir de vîrfuri $v_1, v_2, \ldots, v_n$ astfel încît v_{j+1} este descendent direct al vîrfului v_j, $j = \overline{1, n-1}$, $v' = v_1$ și $v_n = v''$.

Dacă (v_0, v_1) $(v_1, v_2), \ldots, (v_{n-1}, v_n)$ este drumul care unește vîrful $v = v_n$ cu rădăcina v_0 a arborelui atunci n este *înălțimea vîrfului* v.

Observația I.1.3. Orice graf $\mathfrak{G} = (V, \Gamma)$ definește o relație $\rho_{\mathfrak{G}} \subseteq V \times V$ prin $(v_1, v_2) \in \rho_{\mathfrak{G}}$ dacă și numai dacă $v_2 \in \Gamma(v_1)$.

Invers, pentru orice relaţie $\rho \subseteq V \times V$ există un graf $\mathfrak{G}_\rho = (V, \Gamma_\rho)$ care defineşte această relaţie : Γ_ρ este dată de

$$\Gamma_\rho(v) = \{v_1 \mid (v, v_1) \in \rho\}.$$

Definiţia I.1.23. Un *graf marcat* este un 4-uplu $\mathfrak{G} = (V, \Gamma, I, \mu)$ unde (V, Γ) este un graf, I este *mulţimea simbolurilor de marcare*, iar

$$\mu : \{(v_1, v_2) \mid v_2 \in \Gamma(v_1)\} \to I.$$

este *aplicaţia de marcare*.

Exemplul I.1.7. Fie graful marcat $\mathfrak{G} = (V, \Gamma, I, \mu)$ unde (V, Γ) este graful din fig. I.1.1, $I = \{0, 1\}$, $\mu((v_1, v_1)) = 0$, $\mu((v_1, v_2)) = 1$, $\mu((v_2, v_1)) = 0$, $\mu((v_2, v_2)) = 0$, $\mu((v_2, v_3)) = 1$. Acest graf marcat este reprezentat în fig. I.1.3 indicîndu-se pe fiecare arc simbolul de *marcaj* corespunzător.

Noţiunea de graf este generalizată în

Definiţia I.1.24. Un *hipergraf* este o pereche $\mathfrak{H} = (X, E)$ unde X este o mulţime numită mulţimea *vîrfurilor hipergrafului*, iar E este o familie de părţi ale lui X, $E = \{E_j \mid j \in J\}$ cu proprietăţile :

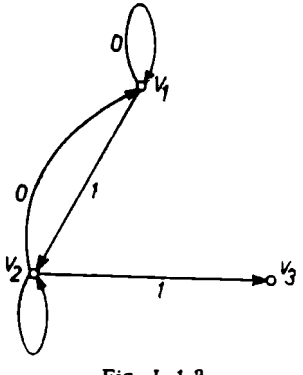

Fig. I. 1.8.

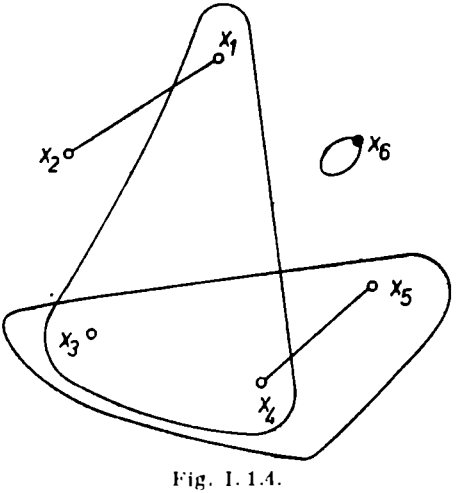

Fig. I. 1.4.

1) $$E_j \neq \emptyset, \ \forall j \in J$$

2) $$\cup \{E_j \mid j \in J\} = X,$$

numite *arce ale hipergrafului*.

Dacă $|X| < \infty$ hipergraful $\mathfrak{H}$ este finit ; în caz contrar $\mathfrak{H}$ este un hipergraf infinit.

Observaţia I.1.4. Oricărui graf $\mathfrak{G} = (V, \Gamma)$ i se poate ataşa un hipergraf $\mathfrak{H} = (V, E)$, arcele hipergrafului fiind mulţimile date de

i) $\{v\}$, dacă $C_v = \{v\}$

ii) $\{v_1, v_2\}$, dacă $v_2 \in \Gamma(v_1)$.

Exemplul I.1.8. Fie hipergraful $\mathfrak{H} = (\{x_1, x_2, x_3, x_4, x_5, x_6\}, \{E_1, E_2, E_3, E_4, E_5\})$ unde $E_1 = \{x_1, x_3, x_4\}$, $E_2 = \{x_3, x_4, x_5\}$, $E_3 = \{x_4, x_5\}$, $E_4 = \{x_6\}$, $E_5 = \{x_1, x_2\}$. Acest hipergraf este reprezentat în figura I.1.4.

Definiţia I.1.25. Un *drum de lungime n într-un hipergraf* $\mathfrak{H} = (X, E)$ este un şir $\{x_1, E_1, x_2, E_2, \ldots, E_n, x_{n+1}\}$ astfel încît :

i) $x_1, \ldots, x_n$ sînt vîrfuri distincte ;

ii) $E_1, \ldots, E_n \in E$ sînt arce distincte ;

iii) $x_k, x_{k+1} \in E_k$ pentru $k = \overline{1, n}$.

Pentru $n > 1$ şi $x_{n+1} = x_1$ obţinem circuite de lungime n.

Observaţia I.1.5. Relaţia $\rho \subseteq X \times X$ dată de $(x_1, x_2) \in \rho$ dacă şi numai dacă există un drum în hipergraful $\mathfrak{H} = (X, E)$ care pleacă din vîrful x_1 la vîrful x_2 este o relaţie de echivalenţă. Clasele de echivalenţă corespunzătoare se numesc *componentele conexe ale hipergrafului*. Hipergraful este conex dacă $|X/\rho| = 1$.

Definiţia I.1.26. *Vîrfurile interioare* ale arcului E_j sînt vîrfurile din mulţimea $E_j \setminus \bigcup \{E_k \mid k \neq j\}$; celelalte vîrfuri ale arcului E_j sînt *vîrfurile conectate* ale arcului.

Definiţia I.1.27. Un *hiperarbore* este un hipergraf conex $\mathfrak{H} = (X, E)$, fără circuite, avînd $|X| \geqslant 2$ şi care satisface la următoarele condiţii :

i) există un vîrf x_0 care nu este vîrf conectat al vreunui arc ; x_0 se numeşte *rădăcina hiperarborelui*.

ii) orice vîrf conectat al unui arc este vîrf conectat pentru exact încă un arc.

Vîrfurile neconectate diferite de x_0 sînt *vîrfurile terminale* ale hiperarborelui.

Exemplul I.1.9. Hiperarborele $\mathfrak{H} = (\{x_0, x_1, x_2, x_3, x_4, x_5, x_6; x_7\}, \{E_1, E_2, E_3, E_4\})$ unde $E_1 = \{x_0, x_1, x_2\}$, $E_2 = \{x_1, x_3, x_4\}$, $E_3 = \{x_3, x_5\}$ şi $E_4 = \{x_4, x_6, x_7\}$ are ca rădăcină vîrful x_0, ca noduri conectate x_1, x_3, x_4, iar ca noduri terminale x_2, x_5, x_6, x_7. Acest hiperarbore este reprezentat în figura I.1.5.

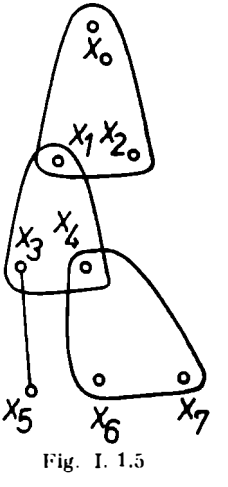

Fig. I. 1.5

Definiţia I.1.28. Vîrful x'' este *descendent al vîrfului* x' în hipergraful $\mathfrak{H} = (X, E)$ dacă există un drum de la x' la x'' ; dacă acest drum are lungimea unu sau dacă $x'' = x'$ atunci x'' este un *descendent imediat* al lui x'. Un descendent x'' al lui x' cu $x'' \neq x'$ este un *descendent propriu*, în caz contrar un *descendent impropriu*.

Ultima parte a acestui paragraf este consacrată prezentării noţiunii de număr cardinal.

Fie T o mulţime, pe care o vom numi în continuare *mulţimea universală*. Vom introduce pe mulţimea $\mathcal{P}(T)$ o relaţie binară

$\gamma = \{(M_1, M_2) \mid M_1, M_2 \in \mathcal{P}(T), \exists f: M_1 \to M_2, f \text{ aplicație bijectivă}\}$.

Teorema I.1.4. *Relația γ este o relație de echivalență.*

Demonstrație. Deoarece aplicația $1_M : M \to M$ este o bijecție urmează $(M, M) \in \gamma$, $\forall M \in \mathcal{P}(T)$ adică γ este o relație reflexivă.

Dacă $(M_1, M_2) \in \gamma$ există bijecția $f: M_1 \to M_2$. Atunci există bijecția $f^{-1}: M_2 \to M_1$ și deci $(M_2, M_1) \in \gamma$, adică γ este o relație simetrică.

Fie $(M_1, M_2), (M_2, M_3) \in \gamma$. Există bijecțiile $f: M_1 \to M_2$ și $g: M_2 \to M_3$. Deoarece aplicația $gf: M_1 \to M_3$ este și ea o bijecție avem $(M_1, M_3) \in \gamma$ deci γ este tranzitivă.

Definiția I.1.29. Un *număr cardinal* este o clasă de echivalență din mulțimea $\mathcal{P}(T)/\gamma$. *Numărul cardinal al mulțimii* M este numărul cardinal căruia îi aparține M. Vom nota acest număr cu $|M|$.

Definiția I.1.30. O *mulțime infinit numărabilă* este o mulțime M pentru care $(M, N) \in \gamma$, unde N este mulțimea numerelor naturale. O *mulțime finită* este o mulțime M pentru care există $n \in N$ astfel ca $(M, \{1,...,n\}) \in \gamma$. Notăm în acest caz $|M| = n$ și convenim că $|\varnothing| = 0$.

O *mulțime numărabilă* este o mulțime finită sau infinit numărabilă.

Observația I.1.6. Orice parte a unei mulțimi numărabile este o mulțime numărabilă.

Exemplul I.1.10. Mulțimea numerelor pare P este infinit numărabilă deoarece aplicația $f: N^* \to P$ dată de $f(n) = 2n$ este o bijecție.

Exemplul I.1.11. Mulțimea perechilor ordonate de numere naturale $M = \{(m, n) \mid m, n \in N^*\}$ este infinit numărabilă. Fie $M_k = \{(m, n) \mid m + n = k + 1\}$; $M = \bigcup_{k=1}^{\infty} M_k$. Mulțimea M_k conține k elemente deoarece m poate lua valori de la 1 la k. Astfel aplicația $\beta: M \to N$ dată de $\beta((m, n)) = \frac{(m + n - 2)(m + n - 1)}{2} + m$ este o aplicație bijectivă. Ea indică locul prerechii (m, n) în șirul

$$(1,1)\ (1,2),\ (2,1),\ (1,3),\ (2,2),\ (3,1),\ \ldots$$

în care apar succesiv elementele mulțimilor M_k și în care perechile fiecărei mulțimi M_k sînt plasate în ordinea crescătoare a primei lor componente.

Să notăm $m + n - 1 = A$. Vom arăta că există un număr A, unic, cu proprietatea:

$$\text{(I.1.1)} \qquad \frac{A(A-1)}{2} < h \leqslant \frac{A(A+1)}{2}.$$

Din inegalitățile (I.1.1) obținem:

$$\frac{\sqrt{1 + 8h} - 1}{2} < A \leqslant \frac{\sqrt{1 + 8h} + 1}{2},$$

și cum intervalul $\left(\frac{\sqrt{1+8h}-1}{2}, \frac{\sqrt{1+8h}+1}{2}\right)$ are lungimea 1 el nu conține decît un singur număr natural.

Astfel $\beta((m, n)) = \frac{A(A-1)}{2} + m$. Oricare ar fi $h \in N^*$ în baza celor arătate mai sus se poate determina A în mod unic astfel ca $\beta((m, n)) = h$, deoarece în acest caz

$$\frac{A(A-1)}{2} < h \leqslant \frac{A(A+1)}{2}.$$

Astfel $m = h - \frac{A(A-1)}{2}$ și $n = A - m + 1$, de unde rezultă și unicitatea perechii (m, n) cu proprietatea $\beta((m, n)) = h$. Conchidem că aplicația β este bijectivă și astfel aplicația $\beta^{-1} : N^* \to M$ realizează „numărarea" elementelor mulțimii M.

Rezultă de aici că dacă M este o mulțime infinit numărabilă atunci mulțimea $M^2 = M \times M$ este infinit numărabilă.

Exemplul I.1.12. Mulțimea numerelor raționale Q este infinit numărabilă deoarece ea poate fi pusă în corespondență biunivocă cu mulțimea perechilor ordonate de numere naturale. Astfel fie r un număr rațional. Există în mod unic perechea de numere naturale (p, q), prime între ele astfel încît $r = p/q$. Aplicația $\gamma : Q \to N_1$ cu $N_1 \subseteq N \times N$, N_1 fiind format din acele perechi de numere naturale, prime între ele, dată de $\gamma(r) = \gamma(p/q) = (p, q)$ este o bijecție.

Nu orice mulțime este numărabilă, așa cum rezultă din

Exemplul I.1.13. Mulțimea numerelor reale din intervalul $[0, 1]$ nu este numărabilă. Într-adevăr, să presupunem că ea ar fi numărabilă și fie $\{r_j \mid j \in N\} = [0, 1]$ o numerotare a elementelor mulțimii $[0, 1]$.

Orice $r \in [0, 1]$ se poate scrie $r = 0, \alpha_1, \alpha_2, \ldots, \alpha_n \ldots$ cu $\alpha_n \in \{0, 1, \ldots, 9\}$. Să notăm $r(m) = \alpha_m$, $m \in N^*$ și să considerăm numărul real

$$r' = 0, \beta_1, \beta_2 \ldots \beta_n \ldots$$

cu $\beta_j \neq r_j(j)$. Evident $r' \in [0, 1]$ și oricare ar fi j, $r' \neq r_j$. Astfel șirul $\{r_j \mid j \in N\}$ nu epuizează mulțimea $[0, 1]$ așa cum era presupus inițial.

Exemplul I.1.14. Mulțimea de funcții $F = \{f \mid f : N \to \{0, 1\}\}$ nu este numărabilă avînd același număr cardinal cu mulțimea $[0, 1]$. Fie $x = 0, x_1, x_2 \ldots \ldots x_n \ldots$ un număr din intervalul $[0, 1]$ scris în baza 2. Avem $x_j \in \{0, 1\}$. Aplicația $\alpha : [0, 1] \to F$ cu $\alpha(x) = f_x$, unde $f_x(n) = x(n)$ este bijectivă deoarece dacă $f_x(n) = f_y(n)$, $\forall n \in N$ avem $x = y$ și pentru orice funcție $f \in F$ există $x \in [0, 1]$ cu $f_x = f$.

Deoarece $F \subseteq \{f \mid f : N \to N\}$ urmează că nici mulțimea $\{f \mid f : N \to N\}$ nu este numărabilă.

Teorema I.1.5. *Fie $\{M_j \mid j \in N\}$ o mulțime numărabilă de mulțimi numărabile. Atunci $M = \cup \{M_j \mid j \in N\}$ este mulțime numărabilă.*

Demonstrație. Deoarece mulțimea M_j este numărabilă putem scrie $M_j = \{r_1^j, r_2^j, ..., r_n^j, ...\}$ și astfel M_j este pusă în corespondență biunivocă cu mulțimea de perechi $\{(j, 1), (j, 2), ..., (j, n), ...\}$. În consecință mulțimea M este în corespondență biunivocă cu mulțimea perechilor ordonate de numere naturale adică ea este numărabilă [v. exemplul I.1.11]. ■

Teorema I.1.6. *Mulțimea părților finite a unei mulțimi numărabile este o mulțime numărabilă.*

Demonstrație. Fie M o mulțime numărabilă. Să arătăm mai întîi că mulțimea M^k este numărabilă, procedînd prin inducție după k. Pentru $k \leqslant 2$ afirmația este adevărată (v. și exemplul I.1.11).

Să presupunem că M^k este o mulțime numărabilă. Atunci cum M^{k+1} este în corespondență biunivocă cu mulțimea $M^k \times M$ prin aplicația $\alpha : M^{k+1} \to M^k \times M$ dată de

$$\alpha((m_1, ..., m_{k+1})) = ((m_1, ..., m_k), m_{k+1})$$

rezultă că M^{k+1} este numărabilă.

În baza teoremei I.1.5 mulțimea $M_1 = \cup \{M^k \mid k \in N\}$ este o mulțime numărabilă. Mulțimea părților finite ale mulțimii M este în corespondență biunivocă cu mulțimea cît M_1 / ρ unde $\rho = \{((m_1, ..., m_k), (m_1', ..., m_k')) \mid \{m_1, ..., m_k\} = \{m_1, ..., m_k\}\}$. ■

§ I. 2. LATICI

Definiția I.2.1. O *latice* este o mulțime M împreună cu două operații binare pe M, reuniunea și intersecția notate respectiv cu $\cup$ și $\cap$ care se bucură de următoarele proprietăți :

1) $m_1 \cup m_2 = m_2 \cup m_1$, $m_1 \cap m_2 = m_2 \cap m_1$, $\forall m_1, m_2 \in M$ (comutativitate),

2) $m_1 \cup (m_2 \cup m_3) = (m_1 \cup m_2) \cup m_3$, $m_1 \cap (m_2 \cap m_3) = (m_1 \cap m_2) \cap m_3$, $\forall m_1, m_2, m_3 \in M$ (asociativitate),

3) $m_1 \cup (m_2 \cap m_1) = m_1$, $m_1 \cap (m_2 \cup m_1) = m_1$, $\forall m_1, m_2 \in M$ (absorbție).

Vom nota laticea astfel definită cu $(M, \cup, \cap)$.

Observația I.2.1. Remarcăm că în sistemul de axiome din definiția precedentă putem schimba semnele $\cup$ și $\cap$ între ele. Ca atare, acest lucru se poate face în orice afirmație dedusă din acest sistem de axiome, proprietate cunoscută sub numele de *principiul dualității pentru latici.*

Exemplul I.2.1. Mulțimea $\mathcal{P}(M)$ a părților unei mulțimi M împreună cu reuniunea și intersecția formează o latice.

Mulțimea numerelor naturale cu operațiile

$$n_1 \cup n_2 = [n_1, n_2],$$
$$n_1 \cap n_2 = (n_1, n_2),$$

unde $[n_1, n_2]$ și (n_1, n_2) înseamnă respectiv cel mai mic multiplu comun și cel mai mare divizor comun al numerelor n_1 și n_2 formează o latice.

Observația I.2.2. Avem

$m \cup m = m$ și $m \cap m = m$, $\forall m \in M$ (proprietate de idempotență).

Într-adevăr, folosind de două ori proprietatea de absorbție găsim mai întîi $m \cup (m \cap m) = m$, de unde $m \cap [m \cup (m \cap m)] = m \cap m$ sau $m = m \cap m$. Similar se arată că $m = m \cup m$.

Teorema I.2.1. *Fie* $(M, \cup, \cap)$ *o latice. Următoarele două afirmații sînt echivalente:*

$$i)\ m_2 = m_1 \cup m_2,$$
$$ii)\ m_1 = m_1 \cap m_2.$$

Demonstrație. Dacă $m_2 = m_1 \cup m_2$ din proprietatea de absorbție găsim $m_1 = m_1 \cap (m_1 \cup m_2) = m_1 \cap m_2$, adică $i) \Rightarrow ii)$. Implicația $ii) \Rightarrow i)$ se arată analog.

Teorema I.2.2. *Relația*

$$\leqslant = \{(m_1, m_2) \mid m_1 = m_1 \cap m_2\},$$

este o relație de ordine parțială. Mai mult $m_1 \cup m_2 = \sup\{m_1, m_2\}$ *și* $m_1 \cap m_2 = \inf\{m_1, m_2\}$.

Demonstrație. Relația $\leqslant$ este reflexivă (deoarece $m = m \cap m$), antisimetrică (căci $m_1 = m_1 \cap m_2$ și $m_2 = m_2 \cap m_1$ implică $m_1 = m_2$) și tranzitivă (pentru că $m_1 = m_1 \cap m_2$ și $m_2 = m_2 \cap m_3$ atrage $m_1 = m_1 \cap (m_2 \cap m_3) = (m_1 \cap m_2) \cap m_3 = m_1 \cap m_3$).

Avem $m_1 \cup m_2 = \{m_1, m_2\}^s \cap (\{m_1, m_2\}^s)^t$. Într-adevăr $m_1 \leqslant m_1 \cup m_2$ deoarece $m_1 = m_1 \cap (m_1 \cup m_2)$ și similar $m_2 \leqslant m_1 \cup m_2$. Apoi, dacă $m_1 \leqslant m$ și $m_2 \leqslant m$ obținem $m = m_1 \cup m$ și $m = m_2 \cup m$ în baza teoremei I.2.1, de unde $m = (m_1 \cup m) \cup (m_2 \cup m) = (m_1 \cup m_2) \cup m$ sau $m_1 \cup m_2 \leqslant m$, adică $m_1 \cup m_2 \in (\{m_1, m_2\}^s)^t$. Asemănător obținem $m_1 \cap m_2 = \inf\{m_1, m_2\}$. ■

Teorema I.2.3. *Fie* $(M, \leqslant)$ *o mulțime parțial ordonată. Să presupunem că oricare ar fi* $m_1, m_2 \in M$ *există* $\sup\{m_1, m_2\}$ *și* $\inf\{m_1, m_2\}$. *Punînd*

$$m_1 \cup m_2 = \sup\{m_1, m_2\},$$
$$m_1 \cap m_2 = \inf\{m_1, m_2\},$$

mulțimea M cu operațiile $\cup$ *și* $\cap$ *este o latice.*

Demonstrație. Vom verifica axiomele laticelor din definiția I.2.1.

Evident $\sup\{m_1, m_2\} = \sup\{m_2, m_1\}$, de unde $m_1 \cup m_2 = m_2 \cup m_1$. Apoi $\inf\{m_1, m_2\} = \inf\{m_2, m_1\}$ de unde $m_1 \cap m_2 = m_2 \cap m_1$.

Să notăm

$$m' = \sup\{m_1, \sup\{m_2, m_3\}\} \text{ și } m'' = \sup\{\sup\{m_1, m_2\}, m_3\}.$$

Avem $m' \geqslant m_1$ și $m' \geqslant \sup\{m_2, m_3\}$ sau $m' \geqslant m_1, m_2, m_3$. De aici $m' \geqslant \sup\{m_1, m_2\}$ și deci $m' \geqslant m''$. Similar obținem $m'' \geqslant m'$ de unde $m'' = m'$ și rezultă asociativitatea reuniunii.

Demonstrarea asociativității intersecției este asemănătoare.

În sfîrșit, fie $m = \sup\{m_1, \inf\{m_2, m_1\}\}$. Atunci $m \geqslant m_1$. Cum $m_1 \geqslant m_1$ și $m_1 \geqslant \inf\{m_2, m_1\}$ avem $m_1 \geqslant \sup\{m_1, \inf\{m_2, m_1\}\} = m$. Astfel $m = m_1$, adică obținem $m_1 \cup (m_2 \cap m_1) = m_1$. La fel se arată că $m_1 \cap (m_2 \cup m_1) = m_1$ și obținem proprietatea de absorbție. ■

Consecința I.2.1. Fie M o latice față de operațiile $\cup$ și $\cap$ și relația de ordine parțială definită în teorema I.2.2. Operațiile obținute în teorema I.2.3 coincid cu operațiile inițiale.

Astfel o latice poate fi definită fie cu ajutorul operațiilor de reuniune și intersecție, fie cu ajutorul unei relații de ordine parțială pentru care există $\sup\{m_1, m_2\}$ și $\inf\{m_1, m_2\}$ pentru orice pereche de elemente $m_1, m_2 \in M$.

Definiția I.2.2. O *latice completă* este o mulțime parțial ordonată $(M, \leqslant)$ pentru care există $\sup M_1$ și $\inf M_1$, oricare ar fi $M_1 \subseteq M$. Vom nota $\sup M_1 = \bigcup \{m \mid m \in M_1\}$ și $\inf M_1 = \bigcap \{m_1 \mid m_1 \in M_1\}$.

Evident, orice latice completă este o latice, dar nu orice latice este completă.

Exemplul I.2.2. Mulțimea $\mathcal{P}(M)$ a părților unei mulțimi M este o latice completă față de relația de incluziune dintre mulțimi.

În schimb mulțimea $(0, 1)$ este o latice față de relația de ordine obișnuită dintre numerele reale, dar nu este o latice completă deoarece $\sup(0, 1) = 1 \notin (0,1)$ și $\inf(0, 1) = 0 \notin (0, 1)$.

Teorema I.2.4. *Mulțimea parțial ordonată $(M, \leqslant)$ este o latice completă dacă una din următoarele două afirmații este adevărată:*

1) *există* $\inf M_1$, $\forall M_1 \subseteq M$ *și mulțimea* M *conține un cel mai mare element;*

2) *există* $\sup M_1$, $\forall M_1 \subseteq M$ *și mulțimea* M *conține un cel mai mic element.*

Demonstrație. Vom arăta că, în primul caz, există totdeauna $\sup M_1$, $\forall M_1 \subseteq M$. Deoarece mulțimea M conține un cel mai mare element, mulțimea $M_2 = \{m \mid m \in M, m \geqslant m_1, \forall m_1 \in M_1\}$ este nevidă. Fie $m' = \inf M_2$. Să arătăm că $m' = \sup M_1$.

Dacă $m_1 \in M_1$ atunci oricare ar fi $m \in M_2$ avem $m \geqslant m_1$, adică $m_1 \in M_1^{\downarrow}$. Ca atare $\inf M_2 = m' \geqslant m_1$, $\forall m_1 \in M_1$.

Apoi, fie $m'' \geqslant m_1$, $\forall m_1 \in M_1$; atunci $m'' \in M_2$ și astfel $m'' \geqslant \inf M_2$. Conchidem că $\inf M_2 = \sup M_1$.

Similar se arată că și condiția a doua este suficientă pentru ca mulțimea parțial ordonată $(M, \leqslant)$ să fie o latice completă.

Consecința I.2.2. Un sistem de închidere $\mathbf{C} \subseteq \mathcal{P}(M)$ este o latice completă fiind închis la intersecții oarecare și conținînd mulțimea M.

Exemplul I.2.3. Mulțimea $\mathrm{Eq}(M)$ este o latice completă în baza consecinței I.2.2. Atragem atenția asupra faptului că reuniunea în această latice nu coincide cu reuniunea relațiilor considerate ca părți ale mulțimii $M \times M$. În general, dacă E_1 este o mulțime de echivalențe $\sup E_1 \subseteq \bigcup \{\alpha \mid \alpha \in E_1\}$.

Definiția I.2.3. O *latice mărginită* este o latice M care posedă un cel mai mic și un cel mai mare element ; le vom nota respectiv cu 0 și 1. Avem. $\forall m \in M$;

$$0 \leqslant m \leqslant 1,$$

$$0 \cup m = 1 \cap m = m,$$

$$0 \cap m = 0 ; \ 1 \cup m = 1.$$

Observația I.2.3. O latice completă este mărginită luînd 1 ca intersecția familiei vide de mulțimi, iar 0 ca reuniunea aceleiași familii.

Teorema I.2.5. (*teorema de punct fix a lui Knäster-Tarski*). *Fie* $(M, \leqslant)$ *o latice completă și* $f: M \to M$ *o aplicație izotonă* ($\forall m_1, m_2 \in M$, $m_1 \leqslant m_2$ *avem* $f(m_1) \leqslant f(m_2)$). *Există* $m \in M$ *cu* $f(m) = m$.

Demonstrație. Să considerăm mulțimea $M_1 = \{m_1 \mid m_1 \in M, m_1 \leqslant f(m_1)\}$. Avem $M_1 \neq \emptyset$ deoarece $0 \in M_1$. Fie $m = \sup M_1$. Dacă $m_1 \in M_1$ atunci $m_1 \leqslant m$, de unde $m_1 \leqslant f(m_1) \leqslant f(m)$. Urmează de aici că $m = \sup M_1 \leqslant f(m)$ adică $m \in M_1$.

Prin urmare $f(m) \leqslant f(f(m))$, ceea ce arată că $f(m) \in M_1$. În baza definiției lui m avem atunci $f(m) \leqslant m$, de unde $f(m) = m$. Astfel m este un punct fix al aplicației considerate. ■

Definiția I.2.4. Fie $(M, \cup, \cap)$ o latice mărginită și $m \in M$. Un *element complementar* (pe scurt un *complement*) al elementului m este elementul m' cu $m \vee m' = 1$ și $m \wedge m' = 0$.

Observația I.2.4. Nu orice element dintr-o latice mărginită are un complement. Astfel, în laticea mărginită $(\{0, m, 1\}, \cup, \cap)$ elementul m nu are complement.

Complementul unui element, dacă există, nu este în mod necesar unic. Astfel în laticea $(\{0, m_1, m_2, m_3, 1\}, \cup, \cap)$ în care operațiile sînt date în tabloul de mai jos :

$\cup$	0	m_1	m_2	m_3	1
0	0	m_1	m_2	m_3	1
m_1	m_1	m_1	m_2	1	1
m_2	m_2	m_2	m_2	1	1
m_3	m_3	1	1	m_3	1
1	1	1	1	1	1

$\cap$	0	m_1	m_2	m_3	1
0	0	0	0	0	0
m_1	0	m_1	m_1	0	m_1
m_2	0	m_1	m_2	0	m_2
m_3	0	0	0	m_3	m_3
1	0	m_1	m_2	m_3	1

elementul m_3 are două complemente: pe m_1 și pe m_2.

Remarcăm că elementele 0 și 1 au fiecare complement unic, respectiv pe 1 și 0.

Definiția I.2.5. O *latice complementară* este o latice mărginită în care orice element m are un complement unic m^c.

Definiția I.2.6. O *latice distributivă* $(M, \cup, \cap)$ este o latice în care $\forall m_1, m_2, m_3 \in M$ avem:

$$(m_1 \cup m_2) \cap m_3 = (m_1 \cap m_3) \cup (m_2 \cap m_3). \tag{I.2.1}$$

Observația I.2.5. Într-o latice distributivă are loc și egalitatea

$$(m_1 \cap m_2) \cup m_3 = (m_1 \cup m_3) \cap (m_2 \cup m_3),$$

care se obține din (I.2.1) aplicînd principiul dualității.

Teorema I.2.6. *Într-o latice distributivă mărginită complementul unui element, dacă există, este unic.*

Demonstrație. Să presupunem că

$$m \cap m' = m \cap m'' = 0,$$

$$m \cup m' = m \cup m'' = 1.$$

Atunci $m' = m' \cap 1 = m' \cap (m \cup m'') = (m' \cap m) \cup (m' \cap m'') = m' \cap m''$ adică $m' \geqslant m''$. La fel se arată că $m'' \geqslant m'$ de unde $m' = m''$. ■

Definiția I.2.7. O *algebră Boole* este o latice distributivă și complementară.

În baza teoremei I.2.6 într-o algebră Boole complementul unui element m este unic și va fi notat cu m^c. Se poate concepe o algebră Boole ca un 4-uplu $(M, \cup, \cap, ^c)$ în care ^{c} este operația unară de complimentariere.

Exemplul I.2.4. Mulțimea $(\mathcal{P}(M), \cup, \cap, ^c)$ formează o algebră Boole în care $M_1^c = M \setminus M_1$.

Exemplul I.2.5. Algebra Boole cu două elemente este $B_2 = (\{0, 1\}, \cup, \cap, ^c)$ în care operațiile sînt date mai jos:

$\cup$	0	1
0	0	1
1	1	1

$\cap$	0	1
0	0	0
1	0	1

^{c}	
0	1
1	0

Ea poate servi la construcția corpului finit cu două elemente, notat cu $GF(2)$. Mai precis, considerînd operațiile de adunare $x \oplus y = (x \cap y^c) \cup (x^c \cap y)$ și de înmulțire $xy = x \cap y$ pe mulțimea $\{0, 1\}$ obținem un corp.

Teorema I.2.7. *Într-o algebră Boole au loc relațiile lui De Morgan*

$$(m_1 \cup m_2)^c = m_1^c \cap m_2^c, \tag{I.2.2}$$

$$(m_1 \cap m_2)^c = m_1^c \cup m_2^c. \tag{I.2.3}$$

Demonstrație. Vom dovedi că $(m_1 \cup m_2) \cup (m_1^c \cap m_2^c) = 1$ și $(m_1 \cup m_2) \cap (m_1^c \cap m_2^c) = 0$ de unde, ținînd seama de unicitatea complementului într-o latice distributivă vom obține formula (I.2.2). Avem

$$(m_1 \cup m_2) \cup (m_1^c \cap m_2^c) = ((m_1 \cup m_2) \cup m_1^c) \cap ((m_1 \cup m_2) \cup m_2^c) =$$
$$= 1 \cap 1 = 1,$$

iar

$$(m_1 \cup m_2) \cap (m_1^c \cap m_2^c) = (m_1 \cap (m_1^c \cap m_2^c)) \cup (m_2 \cap (m_1^c \cap m_2^c)) =$$
$$= 0 \cup 0 = 0.$$

Similar se procedează pentru (I.2.3). ■

Noțiunea de latice se poate generaliza. Astfel are loc

Definiția I.2.8. O *semilatice* este o mulțime M prevăzută cu o operație „$*$" care satisface următoarele axiome :

1) $\forall m_1, m_2 \in M, \quad m_1 * m_2 = m_2 * m_1,$

2) $\forall m_1, m_2, m_3 \in M, \quad m_1 * (m_2 * m_3) = (m_1 * m_2) * m_3,$

3) $\forall m \in M, \quad m * m = m.$

Observația I.2.6. Orice latice $(M, \cup, \cap)$ este semilatice față de fiecare din cele două operații.

§ I. 3. SEMIGRUPURI

Definiția I.3.1. Un *semigrup* este un cuplu $(M, *)$ în care M este o mulțime nevidă, iar „$*$" este o operație binară asociativă ; adică $\forall m_1, m_2, m_3 \in M$ are loc egalitatea :

$$m_1 * (m_2 * m_3) = (m_1 * m_2) * m_3.$$

Un *semigrup cu unitate* (*monoid*) este un triplet $(M, *, e)$ în care $(M, *)$ este un semigrup, iar $e \in M$ are proprietatea :

$$e * m = m * e = m, \ \forall m \in M,$$

și este numit *unitatea semigrupului.*

Exemplul I.3.1. Mulțimea numerelor pare formează semigrup față de operația de înmulțire.

Tripletul $(N, +, 0)$ este un monoid.

Exemplul I.3.2. Orice semilatice $(M, \cup)$ este un semigrup numit *semigrup semilaticial.* Dacă această semilatice are prim element atunci $(M, \cup, 0)$ este *monoid semilaticial.*

Definiţia I.3.2. O *congruenţă la dreapta* (*stînga*) pe semigrupul $(M, *)$ este o echivalenţă $\rho \subseteq M \times M$ cu proprietatea: $(m_1, m_2) \in \rho \Rightarrow (m_1 * m, m_2 * m) \in \rho$ $(\Rightarrow (m * m_1, m * m_2) \in \rho)$, $\forall m \in M$.

O *congruenţă* este o echivalenţă care este simultan congruenţă la dreapta şi la stînga.

Observaţia I.3.1. Intersecţia unei familii oarecare de congruenţe (congruenţe la stînga sau la dreapta) este o congruenţă (congruenţă la stînga sau la dreapta).

Astfel mulţimea congruenţelor (congruenţelor la stînga, la dreapta) formează un sistem de închidere.

Fie I o mulţime cel mult numărabilă numită *alfabet.* O *literă* (*un simbol*) este un element oarecare $i \in I$.

Definiţia I.3.3. Un cuvînt peste alfabetul I este o aplicaţie $p: \{1,..., n\} \to I$. Aplicaţia p se notează de obicei prin $p = i_1 \dots i_n$, unde $i_j = p(j)$, $j = \overline{1, n}$.

Lungimea cuvîntului p este $l(p) = l(i_1 \dots i_n) = n$, adică numărul simbolilor din I care apar în cuvîntul p.

În mulţimea cuvintelor se introduce operaţia de concatenare (juxtapunere). Fie p şi q două cuvinte peste acelaşi alfabet I, $p = i_1 \dots i_m$, $q = i'_1 \dots i'_n$.

Definiţia I.3.4. *Cuvîntul obţinut prin concatenare* din cuvintele p şi q este cuvîntul $pq = i_1 \dots i_m i'_1 \dots i'_n$.

Avem $l(pq) = l(p) + l(q)$.

Observaţia I.3.2. Mulţimea tuturor cuvintelor peste un alfabet I, dotată cu operaţia de concatenare este un semigrup numit *semigrupul liber* generat de I, notat cu I^*.

În semigrupul liber I^* există elementul neutru la operaţia de concatenare: *cuvîntul nul* e, cu $l(e) = 0$. Avem $ep = pe = p$, $\forall p \in I^*$.

Denumirea de „semigrup liber" este justificată prin:

Teorema I.3.1. *O aplicaţie* $\varphi: I \to S$, *unde* S *este un semigrup oarecare se extinde în mod unic la un homomorfism* $\varphi_1: I^* \to S$.

Demonstraţie. Fie $p \in I^*$. Vom defini $\varphi_1(p) = \varphi_1(i_1 \dots i_n) \overset{\text{def}}{=} \varphi(i_1)\varphi(i_2)\dots \dots \varphi(i_n)$. Se constată imediat că φ_1 astfel obţinut este un homomorfism de semigrupuri care coincide pe generatorii semigrupului liber I^* cu aplicaţia φ.

Să presupunem că ar exista un homomorfism $\varphi_2: I^* \to S$ astfel încît $\varphi_2 \restriction I = \varphi_1 \restriction I$.

Mulţimea E definită prin $E = \{p \mid \varphi_1(p) = \varphi_2(p)\}$ este un subsemigrup al lui I^*.

Să observăm mai întîi că $E \neq \emptyset$ deoarece $I \subset E$.

Fie $p, q \in E$. Atunci $\varphi_1(pq) = \varphi_1(p)\varphi_1(q) = \varphi_2(p)\varphi_2(q) = \varphi_2(pq)$, deci $pq \in E$.

Cum E este un semigrup care conţine mulţimea de generatori ai semigrupului I^*, urmează $E = I^*$. ■

Definiţia I.3.5. Un *limbaj* pe alfabetul I este o parte a semigrupului liber I^*.

Mulţimea limbajelor pe alfabetul I, $\mathcal{P}(I^*)$ este o algebră Boole faţă de operaţiile obişnuite din teoria mulţimilor, de reuniune, intersecţie şi complementariere.

Vom utiliza *produsul* dintre două limbaje. Astfel, produsul limbajelor E_1 şi E_2 este limbajul

$$E_1 E_2 = \{p \mid p = p_1 p_2,\ p_1 \in E_1,\ p_2 \in E_2\}.$$

Produsul de limbaje este asociativ, dar nu comutativ. Limbajul $\{e\}$ joacă rolul elementului neutru la produs, $\{e\} E = E \{e\} = E$.

Limbajul vid, $\varnothing$ este element nul faţă de aceeaşi operaţie, $\varnothing E = E\varnothing = \varnothing$.

Închiderea prin produs a limbajului E este limbajul

$$E^* = \bigcup_{k=0}^{\infty} E^k,$$

unde $E^0 = \{e\}$, iar $E^k = E\ E^{k-1}$, $\forall k \in N^*$.

§ I. 4. SPAŢII METRICE ŞI SPAŢII LINEARE

Fie M o mulţime oarecare.

Definiţia I.4.1. O *metrică* pe mulţimea M este o aplicaţie $d : M \times M \to R$ care are următoarele proprietăţi:

1) $d(m_1, m_2) \geqslant 0, \forall m_1, m_2 \in M$ şi $d(m_1, m_2) = 0$ dacă şi numai dacă $m_1 = m_2$;

2) $d(m_1, m_2) = d(m_2, m_1),\ \forall m_1, m_2 \in M$,

3) $d(m_1, m_3) \leqslant d(m_1, m_2) + d(m_2, m_1),\ \forall m_1, m_2, m_3 \in M$.

Perechea (M, d) se numeşte *spaţiu metric*.

Definiţia I.4.2. Un *spaţiu liniar real* sau pe scurt, un *spaţiu liniar* este o mulţime M prevăzută cu două operaţii:

1) o operaţie internă, adunarea faţă de care M formează un grup abelian, adică sînt satisfăcute axiomele :

1a) $m_1 + (m_2 + m_3) = (m_1 + m_2) + m_3,\ \forall m_1, m_2, m_3 \in M$ (asociativitate),

1b) $\exists o \in M$ astfel încît $o + m = m + o = m,\ \forall m \in M$ (existenţa elementului nul),

1c) Pentru $m \in M$, $\exists m' \in M$ cu $m + m' = m' + m = o$. Vom nota $m' = -m$ (existența elementului opus).

1d) $m_1 + m_2 = m_2 + m_1$, $\forall m_1, m_2 \in M$ (comutativitate).

2) o operație externă, înmulțirea cu elementele din R față de care sînt satisfăcute axiomele :

2a) $\alpha(m_1 + m_2) = \alpha m_1 + \alpha m_2$, $\forall \alpha \in R$, $m_1, m_2 \in M$,

2b) $(\alpha + \beta) m = \alpha m + \beta m$, $\forall \alpha, \beta \in R$, $m \in M$,

2c) $\alpha(\beta m) = (\alpha\beta) m$, $\forall \alpha, \beta \in R$, $m \in M$,

2d) $1m = m$, $\forall m \in M$,

Elementele lui M se numesc vectori, elementele lui R scalari, iar cele două operații se numesc respectiv adunare de vectori și înmulțirea cu scalari.

Exemplul I.4.1. Mulțimea numerelor complexe C formează un spațiu liniar față de adunarea numerelor complexe și înmulțirea cu numere reale.

Definiția I.4.3. O *normă* pe spațiul liniar M este o aplicație $\| \cdot \| : M \to R$ care are următoarele proprietăți :

1) $\| m \| \geqslant 0$, $\forall m \in M$ și $\| m \| = 0$ dacă și numai dacă $m = o$,

2) $\| m_1 + m_2 \| \leqslant \| m_1 \| + \| m_2 \|$ $\forall m_1, m_2 \in M$,

3) $\| \alpha m \| = | \alpha | \, \| m \|$, $\forall \alpha \in R$, $m \in M$.

Norma vectorului m este numărul $\| m \|$. Cuplul $(M, \| \cdot \|)$ se numește *spațiu liniar normat.*

Exemplul I.4.2. Aplicația $| \cdot | : C \to R$ dată de $| a + bi | = \sqrt{a^2 + b^2}$ este o normă pe C.

Teorema I.4.1. *Fie* $(M, \| \cdot \|)$ *un spațiu liniar normat. Aplicația* $d : M \times M \to R$ *dată prin* $d(m_1, m_2) = \| m_1 - m_2 \|$ *este o metrică pe* M.

Demonstrație. Avem $d(m_1, m_2) \geqslant 0$ deoarece $\| m_1 - m_2 \| \geqslant 0$. Apoi, $d(m_1, m_2) = 0$ dacă și numai dacă $\| m_1 - m_2 \| = 0$, adică dacă și numai dacă $m_1 - m_2 = o$ sau $m_1 = m_2$.

Simetria metricii se obține imediat : $d(m_1, m_2) = \| m_1 - m_2 \| = \| (-1) \cdot (m_2 - m_1) \| = \| m_2 - m_1 \| = d(m_2, m_1)$.

În sfîrșit $d(m_1, m_3) = \| m_1 - m_3 \| = \| m_1 - m_2 + m_2 - m_3 \| \leqslant \| m_1 - m_2 \| + \| m_2 - m_3 \| = d(m_1, m_2) + d(m_2, m_3)$. ■

Teorema precedentă justifică

Definiția I.4.4. *Metrica indusă de norma* $\| \cdot \|$ pe spațiul liniar normat $(M, \| \cdot \|)$ este aplicația $d : M \times M \to R$ dată de $d(m_1, m_2) = \| m_1 - m_2 \|$.

Exemplul I.4.3. Metrica indusă de norma considerată în exemplul I.4.2 este dată de

$$d(a + bi, c + di) = \sqrt{(a - c)^2 + (b - d)^2}.$$

Fie acum (M, d) un spațiu metric.

Definiţia I.4.5. *Şirul* $\{m_n \mid n \in N^*\} \subseteq M$ este *convergent* şi *are limita* $m \in M$ dacă, oricare ar fi $\varepsilon \in R$, $\varepsilon > 0$ există $n_0 \in N$ astfel încît $n \geqslant n_0$ implică $d(m_n, m) < \varepsilon$.

Definiţia I.4.6. Şirul $\{m_n \mid n \in N^*\} \subseteq M$ este *şir Cauchy* dacă oricare ar fi $\varepsilon \in R$, $\varepsilon > 0$ există $n_0 \in N$ astfel încît $n \geqslant n_0$ implică $d(m_n, m_{n+p}) < \varepsilon$, oricare ar fi $p \in N$.

Observaţia I.4.1. Un şir convergent este şir Cauchy. Într-adevăr fie $\{m_n \mid n \in N^*\}$ un şir convergent avînd limita m şi $\varepsilon' = \varepsilon/2$. Există n_0 astfel încît dacă $n \geqslant n_0$ avem $d(m_n, m) < \varepsilon'$, $d(m_{n+p}, m) < \varepsilon'$, $\forall p \in N$. Folosind proprietăţile metricii găsim $d(m_n, m_{n+p}) \leqslant d(m_n, m) + d(m, m_{n+p}) < 2\varepsilon' = \varepsilon$.

Reciproca acestei afirmaţii nu este adevărată aşa cum rezultă din:

Exemplul I.4.4. Fie spaţiul metric $((0, 1], d)$ unde $d(x, y) = x - y$. Şirul $\left\{\frac{1}{n} \mid n \in N^*\right\}$ este un şir Cauchy, deoarece dacă $n > \frac{1}{\varepsilon}$ avem $\frac{1}{n} - \frac{1}{n+p} =$

$$= \frac{p}{n(n+p)} < \frac{p}{\frac{1}{\varepsilon}\left(\frac{1}{\varepsilon} + p\right)} = \frac{p}{p + \frac{1}{\varepsilon}}\,\varepsilon < \varepsilon.$$

În schimb şirul $\left\{\frac{1}{n} \mid n \in N^*\right\}$ nu este şir convergent în cadrul spaţiului metric specificat. Dacă am considera spaţiul metric $([0, 1], d)$ şirul ar deveni convergent deoarece limita sa este $0 \in [0, 1]$.

Apare astfel justificată:

Definiţia I.4.7. Un *spaţiu metric complet* este un spaţiu metric în care orice şir Cauchy este un şir convergent.

Fie (M, d) un spaţiu metric şi $A : M \to M$ o aplicaţie; vom nota cu Am elementul $A(m)$.

Teorema I.4.2. (*teorema de punct fix a lui Banach*). *Dacă* (M, d) *este un spaţiu metric complet şi dacă* $\forall m_1, m_2 \in M$ *avem* $d(Am_1, Am_2) \leqslant a\, d(m_1, m_2)$, *unde* $a < 1$ *atunci există în* M *un element unic* m *astfel încît* $Am = m$; m *este punctul fix al operatorului* A.

Demonstraţie. Fie $m_0 \in M$; formăm şirul

$$m_1 = Am_0,\ m_2 = Am_1,\ \ldots,\ m_{n+1} = Am_n,\ \ldots$$

Avem $d(m_2, m_1) = d(Am_1, Am_0) \leqslant a\, (d(m_1, m_0)$. În general $d(m_{n+1}, m_n) = d(Am_n, Am_{n-1}) \leqslant a^n d(m_1, m_0)$.

Mai departe obţinem

$$d(m_{n+p}, m_n) \leqslant d(m_{n+p}, m_{n+p-1}) + d(m_{n+p-1}, m_{n+p-2}) + \ldots +$$

$$+ d(m_{n+1}, m_n) \leqslant (a^n + a^{n+1} + \ldots + a^{n+p-1})\, d(m_1, m_0) =$$

$$= a^n \frac{1 - a^p}{1 - a}\, d(m_1, m_0) < \frac{a^n}{1 - a}\, d(m_1, m_0).$$

Cum $a < 1$ avem $\lim_{n\to\infty} d(m_{n+p}, m_n) = 0$, adică şirul $\{m_n \mid n \in N\}$ este şir Cauchy. Spaţiul (M, d) fiind complet şirul este convergent şi fie m limita lui. Vom arăta că $Am = m$.

Avem $d(Am, m) \leqslant d(Am, m_n) + d(m_n, m) = d(Am, Am_{n-1}) + d(m_n, m) \leqslant a\, d(m, m_{n-1}) + d(m_n, m)$.

Deoarece $\lim_{n\to\infty} m_n = m$ oricare ar fi $\varepsilon \in R_+$ există $n_0 \in N$ astfel că dacă $n \geqslant n_0$ avem $d(m_n, n) < \frac{1}{2}\varepsilon$ şi $d(m, m_{n-1}) < \frac{1}{2}\varepsilon$. Atunci $d(Am, m) < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$ ceea ce implică $d(Am, m) = 0$ adică $Am = m$.

Punctul fix m al aplicaţiei A este unic. Într-adevăr să presupunem că ar mai exista un punct fix m' al aplicaţiei A. Atunci $d(m, m') = d(Am, Am') \leqslant a\, d(m, m')$, adică $(1 - a)\, d(m, m') \leqslant 0$, ceea ce nu se poate întîmpla decît dacă $d(m, m') = 0$ adică $m = m'$. ■

O aplicaţie $A : M \to M$ cu proprietatea din enunţul teoremei se numeşte *aplicaţie contractantă* sau *operator contractant* pe spaţiul metric M.

Observaţia I.4.2. Teorema lui Banach îşi menţine valabilitatea dacă operatorul A este contractant numai pe mulţimile $A^h(M)$, $h \in N^*$.

§ I. 5. SPAŢIUL METRIC AL LIMBAJELOR

Fie I un alfabet finit. Vom organiza mulţimea $\mathcal{P}(I^*)$ ca un spaţiu metric. În acest scop vom introduce pe $\mathcal{P}(I^*)$ o structură de spaţiu liniar normat şi vom considera apoi metrica indusă de normă.

Teorema I.5.1. *Mulţimea $\mathcal{P}(I^*)$ este un spaţiu liniar peste corpul $GF(2) = \{0, 1\}$, faţă de operaţia internă* :

$$E_1 + E_2 = (E_1 \cap E_2^c) \cup (E_1^c \cap E_2) = (E_1 \cup E_2) \setminus (E_1 \cap E_2)^{+)}$$

şi operaţia externă, de înmulţire cu scalari : $0 \cdot E = E \cdot 0 = \varnothing$ şi $1 \cdot E = E \cdot 1 = E$.

Demonstraţie. Se verifică uşor că $(\mathcal{P}(I^*), +)$ este un grup aditiv abelian în care elementul nul este $\varnothing$, iar opusul lui E este tot E şi că :

$$\alpha(E_1 + E_2) = \alpha E_1 + \alpha E_2,$$

$$(\alpha + \beta) E_1 = \alpha E_1 + \beta E_1,$$

$\forall \alpha, \beta \in GF(2)$, $E_1, E_2 \in \mathcal{P}(I^*)$. ■

+) $E_1 + E_2$ este diferenţa simetrică a mulţimilor E_1 şi E_2.

Teorema 1.5.2. *Aplicația* $\|.\| : \mathcal{P}(I^*) \to R$ *dată de* $\|E\| = 2^{-l(E)}$, *unde* $l(E) = \min\{l(p) \mid p \in E\}$ *este o normă pe* $\mathcal{P}(I^*)$. *Vom lua* $l(\emptyset) = \infty$.

Demonstrație. Vom verifica faptul că aplicația $\|.\|$ satisface proprietățile unei norme.

1) Avem $\|E\| \geqslant 0$, $\forall E \in \mathcal{P}(I^*)$ și $\|E\| = 0$ dacă și numai dacă $E = \emptyset$.

2) $\|\alpha E\| = |\alpha| \, \|E\|$, $\forall \alpha \in GF(2)$ și $E \in \mathcal{P}(I^*)$. Aici $|\alpha|$ ia valorile reale 0 sau 1 după cum α este unul din elementele 0 sau 1 ale corpului $GF(2)$.

3) $\|E_1 + E_2\| \leqslant \|E_1\| + \|E_2\|$.

Pentru a demonstra această relație să observăm că, dacă $E \subseteq E'$ atunci $\|E\| \leqslant \|E'\|$ și $\|E_1 \cup E_2\| = \max\{\|E_1\|, \|E_2\|\}$, deoarece $l(E_1 \cup E_2) = \min\{l(E_1), l(E_2)\}$.

Atunci $\|E_1 + E_2\| \leqslant \|E_1 \cup E_2\| = \max(\|E_1\|, \|E_2\|) \leqslant \|E_1\| + \|E_2\|$. ■

Consecința 1.5.1. Spațiul normat $(\mathcal{P}(I^*), +, \ldots \|.\|)$ este un spațiu metric având metrica:

$$\rho(E_1, E_2) = \|E_1 - E_2\| = \|E_1 + E_2\|.$$

Vom caracteriza în continuare șirurile Cauchy în acest spațiu.

Teorema 1.5.3. *Șirul* $\{E_n \mid n \in N\}$ *este șir Cauchy dacă și numai dacă pentru orice cuvînt* $p \in I^*$ *putem avea numai una din următoarele două situații:*

1) $\exists n(p) \in N$ *astfel încît pentru* $\forall n \geqslant n(p)$ *avem* $p \in E_n$.

2) $\exists n(p) \in N$ *astfel încît pentru* $\forall n \geqslant n(p)$ *avem* $p \notin E_n$.

Demonstrație. Să presupunem că șirul $\{E_n \mid n \in N^*\}$ este șir Cauchy. Atunci $\forall \varepsilon > 0$, $\exists n_\varepsilon \in N$ astfel ca $\|E_{n+m} - E_n\| < \varepsilon$ de îndată ce $n > n_\varepsilon$, $m \in N^*$.

Fie $p \in I^*$ și $\varepsilon \leqslant 2^{-l(p)}$. Atunci p nu poate aparține numai unuia din limbajele E_{n+m} și E_n pentru $n > n_\varepsilon$ deoarece în acest caz am avea $l(E_{n+m} - E_m) = l(E_{n+m} + E_n) \leqslant l(p)$, de unde $\|E_{n+m} - E_n\| \geqslant 2^{-l(p)} \geqslant \varepsilon$. Așadar luînd $n(p) = n_\varepsilon + 1$, p se poate afla numai într-una din cele două situații de mai sus.

Putem să arătăm acum că dacă șirul $\{E_n \mid n \in N^*\}$ are proprietatea din enunț atunci el este șir Cauchy.

Fie $\varepsilon > 0$. Considerăm limbajul $E_\varepsilon = \{p \mid p \in I^*, l(p) \leqslant -\log_2 \varepsilon\}$, care este evident un limbaj finit.

Pentru orice $p \in E_\varepsilon$ există $n(p)$ astfel încît să avem sau $p \in E_n$ sau $p \notin E_n$ pentru $n > n(p)$.

Fie $n_\varepsilon = \max\{n(p) \mid p \in E_\varepsilon\}$. Atunci, pentru $n > n_\varepsilon$, $p \notin E_{n+m} + E_n$, $\forall m \in N$, de unde $l(E_{m+n} + E_n) = l(E_{m+n} - E_n) > -\log_2 \varepsilon$, adică $\|E_{n+m} - E_n\| < \varepsilon$, pentru $n > n_\varepsilon$. ■

Observația 1.5.1. Din teorema precedentă rezultă că nu orice șir din spațiul considerat este șir Cauchy.

Teorema 1.5.4. *Spațiul metric* $\mathcal{P}(I^*)$ *este complet.*

Demonstrație. Fie $\{E_n \mid n \in N\}$ un șir Cauchy oarecare. Vom arăta existența unui limbaj E astfel încît $\lim_{n \to \infty} \rho(E_n, E) = 0$. Astfel, să luăm $E =$

$= \bigcup_{m=1}^{\infty} \bigcap_{h=m}^{\infty} E_h$, limbaj alcătuit din acele cuvinte p care se află în toate limbajele şirului $\{E_n \mid n \in N^*\}$ începînd de la un anumit rang, adică $E = \lim \inf \{E_k \mid k \in N^*\}$.

Fie $\varepsilon > 0$ şi limbajul finit $E = \{p \mid p \in I^*,\ l(p) \leqslant -\log_2 \varepsilon\}$. Ca şi în demonstraţia teoremei precedente se arată că există n_ε astfel încît pentru $n > n_\varepsilon$ avem $p \in E_n$ sau $p \notin E_n$, oricare ar fi $p \in E_\varepsilon$. Mai mult, $p \notin E_n$, pentru $n > n_\varepsilon$ pentru orice $p \in E_\varepsilon \cap E$. Apoi, dacă $p \in E_\varepsilon$ avem $p \notin E - E_n$, $n > n_\varepsilon$ şi deci $l(E - E_n) > -\log_2 \varepsilon$ sau $\| E - E_n \| < \varepsilon$ pentru $n > n_\varepsilon$. ■

Consecinţa I.5.2. Orice şir ascendent de limbaje $\{E_n \mid n \in N^*\}$, $(E_n \subseteq E_{n+1},\ \forall n \in N^*)$ este convergent şi $\lim_n E_n = \bigcup_n E_n$.

Un şir ascendent este şir Cauchy, deoarece dacă $p \in I^*$ atunci sau există $n_p \in N^*$ astfel ca $p \in E_{n_p}$ (şi deci $p \in E_m$, $\forall m,\ m \geqslant n_p$) sau $p \notin \bigcup E_n$.

În baza teoremei I.5.4. limita şirului ascendent $\{E_n \mid n \in N^*\}$ este $E = \bigcup_{m=1}^{\infty} \bigcap_{n=1}^{\infty} E_k = \bigcup_{m=1}^{\infty} E_m$.

Fie $I' = I \bigcup \{y\}$, $y \notin I$, $T \in \mathcal{P}(I'^*)$ şi $E \in \mathcal{P}(I^*)$.

Definiţia I.5.1. *Operatorul* $\mathbf{T} : \mathcal{P}(I^*) \to \mathcal{P}(I^*)$ *indus* de limbajul $T \in \mathcal{P}(I'^*)$ este aplicaţia care asociază fiecărui limbaj $E \in \mathcal{P}(I^*)$ limbajul $\mathbf{T}(E)$ obţinut din E prin înlocuirea în cuvintele lui T a literei y prin toate cuvintele limbajului E.

Exemplul I.5.1. Fie $I = \{i_1, i_2\}$, $E = \{i_1,\ i_1 i_2,\ i_2 i_1\}$, $T = \{\underbrace{i_1 \dots i_1}_{n}\ y\ \underbrace{i_2 \dots i_2}_{n}$ $n \in N\}$. Atunci $\mathbf{T}(E) = \{\underbrace{i_1 \dots i_1}_{n+1}\ \underbrace{i_2 \dots i_2}_{n},\ \underbrace{i_1 \dots i_1}_{n+1}\ \underbrace{i_2 \dots i_2}_{n+1},\ \underbrace{i_1 \dots i_1}_{n}\ i_2 i_1\ \underbrace{i_2 \dots i_2}_{n}$ $\mid n \in N\}$.

Observaţia I.5.2. Operatorul $\mathbf{T} : \mathcal{P}(I^*) \to \mathcal{P}(I^*)$ este izoton, adică $E_1 \subseteq E_2 \Rightarrow \mathbf{T}(E_1) \subseteq \mathbf{T}(E_2)$.

Fie $T \in \mathcal{P}(I'^*)$. Limbajul T se poate scrie sub forma:

$$T = T_y \bigcup T_{iy} \bigcup T_i, \tag{1.5.1}$$

unde $T_y = T \cap \{y\}^*$, $T_i = T \cap I^*$, $T_{iy} = T \setminus \{T_y \bigcup T_i\}$.

Observaţia I.5.3. Pentru orice $E \in \mathcal{P}(I^*)$ avem:

$$\| \mathbf{T}_y(E) \bigcup \mathbf{T}_{iy}(E) \| \leqslant \| E \|, \tag{I.5.2}$$

deoarece prin înlocuirea simbolului y în cuvintele din $T_y \bigcup T_{iy}$ prin cuvinte din E se obţin cuvinte de lungime cel puţin egală cu cea a cuvintelor iniţiale, dacă $e \notin E$. Dacă $e \in E$ atunci $\| E \| = 1$ şi în acest caz are loc de asemenea inegalitatea (I.5.2).

Definiţia I.5.2. Un operator nesingular este un operator $\mathbf{T}$ pentru care $T_y = \varnothing$.

Observaţia I.5.4. Dacă operatorul $\mathbf{T}$ este nesingular atunci $\| \mathbf{T}_y(E) \bigcup \mathbf{T}_{iy}(E) \| = \| \mathbf{T}_{iy}(E) \| \leqslant \| E \|$, $\forall E \in \mathcal{P}(I^*)$.

Mai mult, putem afirma că $\| \mathbf{T}_{ty}(E) \| \leqslant \frac{1}{2} \| E \|$.

Teorema I.5.5. *Un operator nesingular* $\mathbf{T}$ *este contractant.*

Demonstrație. Este suficient să arătăm că

$$\| \mathbf{T}(E_1) + \mathbf{T}(E_2) \| \leqslant \frac{1}{2} \| E_1 + E_2 \|, \quad \forall E_1, E_2 \in \mathcal{P}(I^*). \tag{I.5.3}$$

Să observăm că dacă $\mathbf{T}$ este nesingular atunci:

$$\| \mathbf{T}(E_1) + \mathbf{T}(E_2) \| = \mathbf{T}_{ty}(E_1) + \mathbf{T}_t(E_1) + \mathbf{T}_{ty}(E_2) + \mathbf{T}_t(E_2) = \\ = \mathbf{T}_{ty}(E_1) + \mathbf{T}_{ty}(E_2) \tag{I.5.4}$$

deoarece $T_t(E_1) = T_t(E_2)$ și deci $T_t(E_1) + T_t(E_2) = \varnothing$.

Fie $r \in \mathbf{T}_{ty}(E_1) + \mathbf{T}_{ty}(E_2)$. El se obține în urma înlocuirii într-un cuvînt oarecare $p_0 y p_1 \ldots y p_k \in T_{ty}$, $p_j \in I^*$, $j = \overline{0, k}$ a simbolului y prin cuvintele $q_1, q_2, \ldots, q_k$ respectiv, unde q_j, $j = \overline{1, k}$ aparțin toate sau lui E_1 sau lui E_2, deci $r = p_0 q_1 p_1 \ldots q_k p_k$. Nu putem avea $q_j \in E_1 \cap E_2$, $j = \overline{1, k}$ deoarece în acest caz $r \in \mathbf{T}_{ty}(E_1) \cap \mathbf{T}_{ty}(E_2)$ și deci $r \notin \mathbf{T}_{ty}(E_1) + \mathbf{T}_{ty}(E_2)$. Prin urmare există un indice j, $1 \leqslant j \leqslant k$ astfel încît $q_j \in E_1$, dar $q_j \notin E_2$ sau invers, deci $q_j \in E_1 + E_2$.

Deoarece avem în mod necesar $\max \{l(p_j), j = \overline{0, k}\} \geqslant 1$ urmează că $l(r) \geqslant l(q_j)$, ceea ce înseamnă că pentru orice cuvînt din $T_{ty}(E_1) + T_{ty}(E_2)$ există un cuvînt de lungime mai mică în $E_1 + E_2$ de unde urmează (I.5.3). ■

Observația I.5.5. Pentru un operator $\mathbf{T}$ singular (cu $T_y \neq \varnothing$) nu mai avem în mod necesar $\max \{l(p_j) \mid j = \overline{0, k}\} \geqslant 1$ și în locul inegalității (I.5.3) avem

$$\| \mathbf{T}(E_1) + \mathbf{T}(E_2) \| \leqslant \| E_1 + E_2 \|. \tag{I.5.5}$$

sau încă

$$\| \mathbf{T}(E_1) - \mathbf{T}(E_2) \| \leqslant \| E_1 - E_2 \|. \tag{I.5.6}$$

Teorema I.5.6. *Orice operator* $\mathbf{T}$ *de forma introdusă în definiția* I.5.1 *este continuu.*

Demonstrație. Rezultă imediat din observația precedentă. ■

Consecința I.5.3. Fie $E = \lim_n E_n$ și $E' = \lim_n E'_n$ și $\mathbf{T}(E_n) \subseteq \mathbf{T}(E'_n)$ pentru $n > n_1$. Atunci $\mathbf{T}(E) \subseteq \mathbf{T}(E')$.

Într-adevăr, din continuitatea operatorului $\mathbf{T}$ avem $\lim_n \mathbf{T}(E_n) = \mathbf{T}(E)$ și $\lim_n \mathbf{T}(E'_n) = \mathbf{T}(E')$. Dacă $r \in \mathbf{T}(E)$ atunci există $n_2 \in N^*$ astfel ca $r \in \mathbf{T}(E_n)$, pentru $n > n_2$. Luînd $n_3 = \max(n_1, n_2)$ avem $r \in \mathbf{T}(E'_n)$ pentru $n > n_3$, de unde $r \in \mathbf{T}(E')$.

Ne vom ocupa în continuare de studiul ecuației:

$$E = \mathbf{T}(E) \tag{I.5.7}$$

Existența unei soluții pentru ecuația (I.5.7) rezultă din teorema I.2.5 de punct fix a lui Knäster-Tarski, deoarece $\mathcal{P}(I^*)$ este o latice completă iar $\mathbf{T}$ este un operator izoton.

Structura de spațiu metric și teorema I.4.2 de punct fix a lui Banach ne vor permite să aducem precizări suplimentare.

Teorema I.5.7. *Limbajul* $E = \lim_n \mathbf{T}^n(K)$, *unde* $K \in \mathcal{P}(I^*)$ *și* $\mathbf{T}^n(K) = \mathbf{T}(\mathbf{T}^{n-1}(K))$ $(\mathbf{T}^1(K) = \mathbf{T}(K))$ *este o soluție a ecuației* (I.5.7).

Demonstrație. Dacă operatorul $\mathbf{T}$ este nesingular atunci el este contractant și afirmația este valabilă în baza teoremei de punct fix a lui Banach.

Fie deci $\mathbf{T}$ un operator singular. Dacă operatorul $\mathbf{T}$ este contractant pe mulțimile $\mathbf{T}^h(K)$, $\mathbf{T}^{h+1}(K)$,..., $h \geqslant 1$ atunci teorema este adevărată așa cum rezultă din observația I.4.2. Pentru a verifica proprietatea de contracție pe mulțimile $\mathbf{T}^h(K)$, $\mathbf{T}^{h+1}(K)$,..., $h \geqslant 1$ va fi suficient să arătăm că

$$\| \mathbf{T}^{m+1}(K) \dot{-} \mathbf{T}^{n+1}(K) \| \leqslant \frac{1}{2} \| \mathbf{T}^m(K) \dot{-} \mathbf{T}^n(K) \|. \tag{I.5.8}$$

pentru $m, n \in N^*$, cu $m, n \geqslant h$.

Fie $r \in \mathbf{T}^{m+1}(K) + \mathbf{T}^{n+1}(K)$. Atunci $r = p_0 q_1 p_1 q_2 \ldots q_k p_k$ și există cuvîntul $p_0 y p_1 y \ldots y p_k \in T_y \cup T_{ty}$ din care obținem cuvîntul r prin înlocuirea lui y prin cuvintele $q_1, \ldots, q_k \in \mathbf{T}^m(K) \cup \mathbf{T}^n(K)$. Să observăm că există un j, $1 \leqslant j \leqslant k$ astfel încît pentru cuvîntul q_j să avem sau $q_j \in \mathbf{T}^m(K)$ și $q_j \notin \mathbf{T}^n(K)$ sau $q_j \notin \mathbf{T}^m(K)$ și $q_j \in \mathbf{T}^n(K)$ deoarece în caz contrar $r \notin \mathbf{T}^{m+1}(K) + \mathbf{T}^{n+1}(K)$. Dacă pentru $\forall m, n > h$ și pentru orice cuvînt r există totdeauna un q_j cu $l(r) > l(q_j)$ atunci inegalitatea (I.5.8) este satisfăcută și operatorul $\mathbf{T}$ este contractant pe mulțimile $\mathbf{T}^h(K)$, $\mathbf{T}^{h+1}(K)$,... Vom arăta că și în cazul cînd inegalitatea (I.5.8) nu-i satisfăcută, ecuația (I.5.7) are soluție. În această situație, are loc egalitatea $l(r) = l(q_j)$ de unde urmează $r = q_j$ ceea ce implică $p_0 = p_1 = \ldots = p_k = e$ și $e \in \mathbf{T}^m(K) \cup \mathbf{T}^n(K)$. Dar dacă $e \in \mathbf{T}^l(K)$, cu $l \geqslant 1$ atunci $e \in T$ sau $e \in K$ deoarece operatorul $\mathbf{T}$ este singular. De aici $e \in \mathbf{T}(K)$, și în general $e \in \mathbf{T}^m(K)$, $\forall m \in N^*$ și aceasta implică $\mathbf{T}(K) \subseteq \mathbf{T}^2(K) \subseteq \ldots \subseteq \mathbf{T}^m(K) \subseteq \ldots$, de unde conform consecinței I.5.2 există $\lim_n \mathbf{T}^n(K) = \bigcup_n \mathbf{T}^n(K)$. Cum operatorul $\mathbf{T}$ este continuu această limită este soluția ecuației (I.5.8). ■

Teorema I.5.8. *Fie* $E_0 = \mathbf{T}(\varnothing) \cup \mathbf{T}^2(\varnothing) \cup \ldots \cup \mathbf{T}^n(\varnothing) \cup \ldots$ *Atunci* $E_0 = \mathbf{T}(E_0)$ *și* $E_0 \subseteq E$ *pentru orice* E *soluție a ecuației* (I.5.7).

Demonstrație. Cum $\varnothing \subseteq \mathbf{T}(\varnothing)$ și $\mathbf{T}$ este izoton urmează $\mathbf{T}^{n-1}(\varnothing) \subseteq \mathbf{T}^n(\varnothing)$, de unde $E_0 = \lim_n \mathbf{T}^n(\varnothing)$ și conform teoremei I.5.7, E_0 este o soluție a ecuației (I.5.7).

Fie E o soluție oarecare a ecuației (I.5.7). Avem $\varnothing \subseteq E$ și în baza izotoniei lui $\mathbf{T}$ urmează $\mathbf{T}^n(\varnothing) \subseteq \mathbf{T}^n(E) = E$, $\forall n \in N$. Trecînd la limită obținem $E_0 \subseteq E$. ■

Această teoremă justifică

Definiția I.5.3. *Soluția minimală* a ecuației (I.5.7) este $E_0 = \bigcup_{h=1}^{\infty} \mathbf{T}^h(\varnothing)$.

Dacă $E_0 = I^*$ ecuația (I.5.6) se zice *ecuație universală.*

Teorema I.5.9. *Ecuația* (I.5.7) *are soluție unică dacă și numai dacă ea este o ecuație universală sau operatorul* **T** *este nesingular.*

Demonstrație. Dacă **T** este un operator nesingular atunci el este contractant și soluția este unică conform teoremei de punct fix a lui Banach.

Dacă ecuația (I.5.7) este universală pentru orice soluție E avem $I^* \subseteq E \subseteq I^*$ și deci $E = I^*$.

Reciproc. să presupunem că ecuația (I.5.7) are soluție unică. Vom arăta că dacă ecuația nu-i universală, atunci cu necesitate operatorul **T** este nesingular.

Să presupunem că operatorul **T** ar fi singular, adică $T_y \neq \varnothing$. Atunci există un cuvînt $\underbrace{yy\ldots y}_{k}$, $k > 0$ care aparține lui T_y. Deoarece $E_0 \neq I^*$ există $p \notin E_0$.

Fie $E = \lim_{n} \mathbf{T}^n(\{p, e\})$. Este clar că $p \in \mathbf{T}^n(\{p, e\})$, $\forall n \in N$ și deci $p \in E$. De aici urmează că $E \neq E_0$ și că E este soluție a ecuației, ceea ce contrazice unicitatea soluției. ■

Observația I.5.6. Rezultatele precedente se extind la ecuații de forma:

$$\mathbf{X} = \mathbf{T}(\mathbf{X}) \tag{I.5.9}$$

unde $\mathbf{X} \in (\mathcal{P}(I^*))^m$, iar $\mathbf{T}: (\mathcal{P}(I^*))^m \to (\mathcal{P}(I^*))^m$, adică $\mathbf{X} = (E_1, E_2, \ldots, E_m)$ în care E_j, $j = \overline{1, m}$ sînt limbaje peste alfabetul finit I. Operatorul $\mathbf{T} = (\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_m)$ va fi în acest caz construit cu ajutorul limbajelor T_k, $k = \overline{1, m}$ peste alfabetul $I' = I \cup \{y_1, \ldots, y_n\}$, unde $y_j \notin I$, $j = \overline{1, n}$, astfel: $\mathbf{T}_k(E_1, \ldots, E_n)$ este limbajul obținut prin înlocuirea în cuvintele lui T_k a simbolilor y_j prin cuvintele din limbajele E_j, $j = \overline{1, n}$, respectiv.

Ecuația (I.5.9) se scrie pe componente:

$$\begin{aligned} E_1 &= \mathbf{T}_1(E_1, \ldots, E_n), \\ &\cdot \\ &\cdot \\ &\cdot \\ E_n &= \mathbf{T}_n(E_1, \ldots, E_n). \end{aligned} \tag{I.5.10}$$

Limbajul T_k se poate scrie prin analogie cu formula (I.5.1) sub forma:

$$T_k = T_{ky} \cup T_{kky} \cup T_{kt}$$

unde $T_{ky} = T_k \cap \{y_1, \ldots, y_n\}^*$, $T_{kt} = T_k \cap I^*$, $T_{ky} = T_k \setminus \{T_{ky} \cup T_{kt}\}$.

Definiția operatorului nesingular se extinde în mod natural: **T** este nesingular dacă $T_{ky} = \varnothing$, $\forall k = \overline{1, n}$. De asemenea rămîne valabilă teorema I.5.9 pentru sistemul (I.5.10) făcînd schimbările care se impun.

Metrica pe spațiul $(\mathcal{P}(I^*))^n$ va fi introdusă prin relația

$$\rho((E_1', ..., E_n'), (E_1'', ..., E_n'')) = \max\{\rho(E_j', E_j'') \mid j = \overline{1, n}\}.$$

Dacă componentele $\mathbf{T}_j$ ale operatorului $\mathbf{T}$ sînt operatori contractanți, atunci și $\mathbf{T}$ este operator contractant.

§ 1.6. ECUAȚII PE MULȚIMEA LIMBAJELOR

Fie $I = \{i_1, i_2, ..., i_n\}$ un alfabet fixat și I^* semigrupul liber generat de I. Vom studia aici sistemele de ecuații pe $\mathcal{P}(I^*)$ echipat atît cu structura de algebră Boole, cît și cea de semigrup față de produsul limbajelor.

Definiția I.6.1. Un *polinom* pe $\mathcal{P}(I^*)$ este:

1) un limbaj de forma $\{i_j\}$, $j = \overline{1, n}$ și limbajul vid;

2) orice limbaj obținut din limbajele $E_1, E_2, ..., E_k$ prin aplicarea de un număr finit de ori a operațiilor de reuniune, produs și închidere prin produs.

Numai limbajele obținute prin procedeele de la punctele 1) și 2) sînt polinoame.

Definiția I.6.2. Un *polinom liniar* în limbajele $Z_1, Z_2, ..., Z_t$ este un limbaj Z de forma:

$$Z = E_0 \cup Z_1 E_1 \cup ... \cup Z_t E_t,$$

$\{E_0, E_1, ..., E_t\}$ fiind *mulțimea coeficienților polinomului* Z, $E_j \in \mathcal{P}(I^*)$, $j = \overline{0, t}$.

În studiul unor clase speciale de limbaje intervin sisteme de polinoame de forma:

$$\text{(I.6.1)} \quad \begin{cases} Z_1 = Z_1 E_{11} \cup Z_2 E_{21} \cup ... \cup Z_m E_{m1} \cup B_1 \\ Z_2 = Z_1 E_{12} \cup Z_2 E_{22} \cup ... \cup Z_m E_{m2} \cup B_2 \\ \cdot \\ \cdot \\ \cdot \\ Z_m = Z_1 E_{1m} \cup Z_2 E_{2m} \cup ... \cup Z_m E_{mm} \cup B_m. \end{cases}$$

Definiția I.6.3. O *soluție* a sistemului (I.6.1) este un m-uplu de limbaje $(Z_1, ..., Z_m)$ din $\mathcal{P}(I^*)$ care satisfac sistemul (I.6.1).

Ca și în algebra liniară folosirea matricilor oferă o modalitate de scriere compactă a sistemelor ușurînd manipularea acestora. În acest scop vom considera matrice ale căror elemente sînt limbaje. Prin $C = (C_{hk})$, $C_{hk} \in \mathcal{P}(I^*)$,

$h = \overline{1, n}$, $h = \overline{1, m}$ vom nota o astfel de matrice avînd n linii şi m coloane; se mai spune că matricea C este de tipul (n, m).

În cazul particular $n = 1$ matricea va fi numită *vector-linie*, iar în cazul $m = 1$, *vector-coloană*. Dacă $n = m$ se obţin matricile pătratice.

Fie $C = (C_{hk})$, $D = (D_{hk})$ două matrici de tipul (n, m). Reuniunea lor este matricea $C \cup D = (C_{hk} \cup D_{hk})$ de acelaşi tip.

Dacă $C_{hk} \subseteq D_{hk}$, $h = \overline{1, n}$, $k = \overline{1, m}$ atunci vom scrie $C \subseteq D$. Se constată imediat că relaţia „$\subseteq$" este o relaţie de parţială ordonare în mulţimea matricilor de acelaşi tip.

Mai mult, mulţimea matricilor de un anumit tip este semilatice faţă de reuniune avînd ca prim element *matricea nulă* ($\varnothing$) (exerciţiu !).

Fie $C = (C_{hk})$ o matrice de tip (n, m) şi $D = (D_{kl})$ o matrice de tip (m, p). Produsul lor este matricea $CD = (\bigcup_{k=1}^{m} C_{hk} D_{kl})$ de tip (n, p).

Cititorul va verifica faptul că mulţimea matricilor pătratice de tip (n, n) formează un *semigrup semilaticial* faţă de operaţiile de produs şi reuniune avînd ca element unitate matricea $U = (U_{jk})$, unde $U_{jj} = \{e\}$, $j = \overline{1, n}$ şi $U_{jk} = \varnothing$, $j \neq k$, $j, k = \overline{1, n}$.

Închiderea prin produs a unei matrici pătratice B este matricea:

$$B^* = \bigcup_{j=0}^{\infty} B^j$$

unde $B^0 = U$, $B^{j+1} = B^j \cdot B$, iar

$$(B^j)_{hk} = \bigcup_{(v_1, v_2, \ldots, v_{j-1})} B_{hv_1} B_{v_1 v_2} \ldots B_{v_{j-1} k},$$

unde reuniunea se ia după toate j-1-uplele $(v_1, \ldots, v_{j-1})$ care se pot forma din mulţimea $\{1, 2, \ldots, n\}$..

Un monom $B_{u_1 u_2} B_{u_2 u_3} \ldots B_{u_t u_{t+1}}$ este:

a) *strict-ciclic* dacă indicii $u_1, u_2, \ldots, u_t$ sînt distincţi şi $u_{t+1} = u_1$. În particular monomul B_{uu} (deci orice element de pe diagonala principală a unei matrici pătratice) este ciclic.

b) *k-ciclic*, dacă conţine k monoame strict-ciclice;

c) *ciclic*, dacă există un $k \in N^*$ astfel încît monomul să fie k-ciclic.

Exemplul I.6.1. Monomul $B_{13} B_{32} B_{21}$ este strict-ciclic, monomul $B_{43} B_{32} B_{24} B_{44} B_{12} B_{25} B_{57} B_{74} B_{40}$ este 3-ciclic, deoarece conţine monoamele strict ciclice $B_{43} B_{32} B_{24}$, B_{44} şi $B_{12} B_{25} B_{57} B_{71}$.

Teorema I.6.1. *Orice monom de forma $B_{jh} B_{h1} B_{h2} \ldots B_{h_{r-1} k}$ unde $B_{jh_1}, \ldots, B_{h_{r-1} k}$ sînt elemente dintr-o matrice pătratică B, de tip (n, n) este cel puţin* $\left[\frac{r}{n}\right]$ *ciclic.*

*) Am notat cu $[x]$ partea întreagă a numărului real x.

Demonstrație. Dacă $r < n$ atunci $\left\lfloor \frac{r}{n} \right\rfloor = 0$ și teorema este banal verificată.

Fie deci $r \geqslant n$ și să presupunem că $ln \leqslant r < (l+1)n$, $l \in N^*$. Vom proceda prin inducție după l.

Pentru $l = 1$ avem $n \leqslant r < 2n$ și mulțimea $\{j, h_1, \ldots, h_{r-1}, k\}$ conține $r+1$ elemente dintre care cel mult $n < r+1$ distincte. Prin urmare cel puțin un indice apare de două ori și monomul conține cel puțin un monom strict ciclic.

Să presupunem acum că teorema este valabilă pentru $l = m$, adică pentru $mn \leqslant r < (m+1)n$, un monom cu r factori este cel puțin m-ciclic.

Fie acum $(m+1)n \leqslant r < (m+2)n$. Ultimii n factori din monom conțin cel puțin un monom strict ciclic. Partea inițială rămasă care conține $r - n$ factori, unde $mn \leqslant r - n < (m+1)n$ este conform cu ipoteza inductivă un monom cel puțin m ciclic. Ca atare întregul monom este cel puțin $m + 1 = \left\lfloor \frac{r}{n} \right\rfloor$-ciclic. ■

Definiția I.6.4. O *matrice singulară* este o matrice pătratică de tip (n, n), $B = (B_{jk})$ pentru care există un monom strict ciclic $B_{j_1 j_2} B_{j_2 j_3} \ldots B_{j_k j_1}$ care conține cuvîntul nul e (ceea ce se întîmplă dacă fiecare din factorii monomului conțin pe e); indicii $j_1, j_2, \ldots, j_k$ se zic *ciclici* în raport cu matricea B.

În caz contrar matricea este *nesingulară*.

Observația I.6.1. O matrice pătratică în care cel puțin un element de pe diagonala principală conține cuvîntul nul este singulară.

Observația I.6.2. Orice putere a unei matrici nesingulare este o matrice nesingulară.

Teorema I.6.2. *Fie B o matrice nesingulară de tip (n, n) și $r = mn$, $m \in N^*$. Atunci :*

$$\min \{l(p) \mid p \in (B^r)_{jk}, j, k = \overline{1, n}\} = m.$$

Demonstrație. $(B^r)_{jk}$ este o reuniune de monoame care conțin r factori. Fiecare monom, în baza teoremei I.6.3, este cel puțin $\left\lfloor \frac{r}{n} \right\rfloor = m$-ciclic. Matricea B fiind nesingulară cuvintele din monoamele strict-ciclice au cel puțin lungimea 1, de unde cuvintele din fiecare monom (deci din $(B^r)_{jk}$) au cel puțin lungimea m. ■

Observația I.6.3. Folosind matricile, sistemul (I.6.1) se scrie sub forma

$$Z = ZE \cup B, \tag{1.6.2}$$

unde Z și B sînt vectori linie, iar E este o matrice pătratică :

$$Z = (Z_1, \ldots, Z_m), \quad B = (B_1, \ldots, B_m), \quad E = \begin{pmatrix} E_{11} & \ldots & E_{1m} \\ \vdots & & \vdots \\ E_{m1} & \ldots & E_{mm} \end{pmatrix}.$$

Ne vom ocupa de existența și unicitatea soluțiilor sistemului (I.6.1).

Teorema I.6.3. *Sistemul* (I.6.2) *are soluție unică* Z *dacă matricea* E *este nesingulară și anume* $Z = BE^*$.

Demonstrație. Să verificăm mai întîi că $Z = BE^*$ este o soluție a sistemului (I.6.2). Avem:

$$ZE \cup B = BE^* E \cup B = B(E^* E \cup U) = BE^* = Z.$$

Să arătăm că aceasta este unica soluție a sistemului. Fie Z_0 o soluție oarecare a sistemului, adică $Z_0 = Z_0 E \cup B$.

Obținem succesiv

$$Z_0 = (Z_0 E \cup B) E \cup B = Z_0 E^2 \cup BE \cup B =$$

$$= (Z_0 E \cup B) E^2 \cup BE \cup B = Z_0 E^3 \cup BE^2 \cup BE \cup B =$$

$$\ldots = Z_0 E^{k+1} \cup BE^k \cup \ldots \cup BE \cup B, \forall k \in N.$$

Prin urmare $BE^* \subseteq Z_0$, ceea ce revine la relația de incluziune între elementele corespunzătoare:

$$(BE^*)_j \subseteq (Z_0)_j, \; j = \overline{1, m}. \tag{I.6.3}$$

Să observăm că produsul a două limbaje conține cuvîntul nul dacă și numai dacă fiecare din limbaje conține cuvîntul nul. Dacă reuniunea a două limbaje nu conține cuvîntul nul el nu aparține nici unuia dintre limbaje. Cum E este o matrice nesingulară niciunul din limbajele $(E^k)_{jh}$ unde $k \geqslant m$, $j, h = \overline{1, m}$ nu conțin cuvîntul nul.

Fie acum $p \in (Z_0)_j$ și $l(p) = l$. Alegem $k + 1 = (l + 1)m$ și fie $Z_0 = Z_0 E^{k+1} \cup BE^k \cup \ldots \cup BE \cup B$. Atunci orice cuvînt din orice limbaj nevid $(E^{k+1})_{hj}$, $h = \overline{1, m}$ este de lungime cel puțin $l + 1$ în baza teoremei I.6.2 și deci p neputînd aparține limbajului $(Z_0 E^{k+1})_j$ aparține limbajului $(BE^k \cup \ldots \cup B)_j$, dacă $(Z_0)_j \subseteq (BE^k \cup \ldots \cup B)_j \subseteq (BE^*)_j$. Avem deci

$$Z_0 = BE^*.$$

și soluția sistemului (I.6.2) este unică. ■

Consecința I.6.1. Dacă $e \notin E$ atunci ecuația $Z = ZE \cup B$ are soluție unică $Z = BE^*$.

Demonstrație. Rezultă din teorema I.6.3 pentru $m = 1$. ■

Teorema I.6.4. *Orice soluție a sistemului* (I.6.2) *este de forma* $Z_0 = (B \cup T)E^*$, *unde* T *este un vector linie, care satisface la anumite condiții ce vor fi precizate ulterior.*

Demonstrație. Fie Z_0 o soluție oarecare a sistemului. Urmînd același raționament ca în demonstrarea teoremei precedente obținem $BE^* \subseteq Z_0$, deci există un vector T astfel încît:

$$Z_0 = BE^* \cup T.$$

Ca atare $BE^* \cup T = BE^* E \cup TE \cup B = BE^* \cup TE$, de unde $TE \subseteq BE^* \cup T$ și deci

$$TE^* \subseteq BE^* \cup TE \subseteq BE^* E \cup BE^* \cup T = BE^* \cup T.$$

ș.a.m.d., adică :

$$TE^k \subseteq BE^* \cup T, \ \forall k \in N.$$

ceea ce implică $TE^* \subseteq BE^* \cup T = Z_0$. Astfel $Z_0 = Z_0 \cup TE^* = (B \cup T)E^*$. ■

Observația I.6.4. Soluția minimală a sistemului (I.6.2) este $Z = BE^*$.

Teorema I.6.5. *Un vector linie de forma* $Z_0 = (B \cup T) E^*$ *este o soluție a sistemului* (I.6.2) *dacă și numai dacă* T *satisface inegalitatea* :

$$T \subseteq BE^* \cup TE^* E. \tag{I.6.4}$$

Demonstrație. Să presupunem că vectorul linie $Z_0 = (B \cup T) E^*$ este o soluție a sistemului (I.6.2). Atunci $(B \cup T) E^* = (B \cup T) E^* E \cup B$ sau $BE^* \cup TE^* = BE^* \cup TE^* E$, de unde $TE^* \subseteq BE^* \subseteq TE^* E$, adică $T \cup TE^* E \subseteq BE^* \cup TE^* E$, ceea ce implică (I.6.4).

Reciproc, fie T un vector linie care satisface condiția (I.6.4). Vom arăta că $Z_0 = (B \cup T) E^*$ este o soluție a sistemului (I.6.2). Într-adevăr, avem $Z_0 E \cup B = (B \cup T) E^* \cup B \subseteq BE^* \cup TE^* EE^* \subseteq BE^* \cup TE^* = (B \cup T)E^* = Z_0$ și $Z_0 = (B \cup T) E^* \subseteq (B \cup BE^* \cup TE^* E) E^* = BE^* \cup TE^* EE^* = B \cup BE^* E \cup TE^* E = (BE^* \cup TE^*) E \cup B = [(B \cup T) E^*] E \cup B = Z_0 E \cup B$. Astfel $Z_0 = (B \cup T) E^*$ este o soluție a sistemului. ■

Teorema I.6.6. *Vectorul-linie* T *satisface condiția* (I.6.4) *dacă și numai dacă* $(T)_j = \varnothing$ *pentru orice* j *care nu intră ca indice în vreun monom strict ciclic (format cu elementele matricii* E*) care conține cuvîntul nul, (adică pentru orice indice* j*, neciclic).*

Demonstrație. Să presupunem că $(T)_j = \varnothing$ pentru indicii j neciclici. Atunci, pentru acești indici avem evident

$$\varnothing = (T)_j \subseteq (BE^* \cup TE^* E)_j.$$

Dacă j nu are proprietatea din enunț, atunci există un monom strict ciclic care conține cuvîntul nul e și în care j apare ca indice. Fără a restrînge generalitatea putem presupune monomul de forma $E_{jh_1} \dots E_{h_{k-1}j}$ (Dacă j intra ca indice într-un monom de forma $E_{lh_1} E_{h_1 h_2} E_{h_2 h_3} \dots E_{h_p j} E_{j h_{p+1}} \dots E_{h_s l}$ am fi putut lua monomul $E_{j h_{p+1}} \dots E_{h_s l} E_{l h_1} \dots E_{h_p j}$). În acest caz avem evident $e \in (E^k)_{jj}$ și deci $(T)_j \subseteq (TE^k)_j$, $k \geqslant 1$. Cum $TE^k \subseteq BE^* \cup TE^* E$, pentru $k \geqslant 1$, urmează $(T)_j \subseteq (BE^* \cup TE^* E)_j$. Deci are loc condiția (I.6.4).

Pentru a demonstra afirmația reciprocă este convenabil să utilizăm următoarea „descompunere canonică".

Un vector linie H, de tip $(1, m)$ se poate descompune în *mod canonic în raport cu matricea* E de tip (m, m) sub forma $H = H_1 \bigcup H_2$ unde :

$$(H_1)_j = \begin{cases} (H)_j, \text{ dacă } j \text{ este ciclic în raport cu matricea } E \\ \varnothing, \text{ în caz contrar,} \end{cases}$$

și

$$(H_2)_j = \begin{cases} \varnothing, \text{ dacă } j \text{ este ciclic în raport cu matricea } E \\ (H)_j, \text{ în caz contrar.} \end{cases}$$

Fie T un vector linie care satisface condiția (I.6.4) și $Z_0 = (B \cup T) E^* = BE^* \cup T$ soluția corespunzătoare a sistemului (I.6.2) existentă conform teoremei precedente.

Fie $T = T_1 \cup T_2$, $B = B_1 \cup B_2$, $Z_0 = Z_1 \cup Z_2$ descompunerile canonice ale vectorilor T, B și Z_0 în raport cu matricea E.

Avem :

$$Z_1 \subseteq BE^* \cup T_1 \text{ și } Z_2 \subseteq BE^* \cup T_2 . \tag{I.6.5}$$

Plecînd de la matricea E vom construi două matrici E_1 și E_2 după cum urmează :

$$(E_1)_{hk} = \begin{cases} E_{hk}, \text{ dacă } h \text{ este ciclic, iar } k \text{ nu este} \\ \varnothing, \text{ în caz contrar,} \end{cases} \tag{I.6.6}$$

$$(E_2)_{hk} = \begin{cases} E_{hk}, \text{ dacă } h \text{ și } k \text{ nu sînt ciclici} \\ \varnothing, \text{ în caz contrar.} \end{cases} \tag{I.6.7}$$

Observăm că matricea E_2 este nesingulară.

Avem $Z_1 \cup Z_2 = (Z_1 \cup Z_2) E \cup B$, de unde $Z_2 = Z_2 E_2 \cup Z_1 E_1 \cup B_2$. Într-adevăr, dacă j este indice ciclic $(Z_2)_j = \varnothing$, iar $(Z_2 E_2 \cup Z_1 E_1 \cup B_2)_j = \varnothing$ deoarece $(Z_2 E_2)_j = \bigcup_k (Z_2)_k (E_2)_{kj} = \varnothing$, $Z_1 E_1 = \bigcup_k (Z_1)_k (E_1)_{kj} = \varnothing$ ținînd seama de relațiile (I.6.7) și (I.6.6), iar $(B_2)_j = \varnothing$.

Dacă j nu este ciclic atunci $(Z_2)_j = Z_j$.

Pe de altă parte $(Z_2 E_2 \cup Z_1 E_1 \cup B_2)_j = \bigcup_k (Z_2)_k (E_2)_{kj} \cup \bigcup_k (Z_1)_k (E_1)_{kj} \cup B_j = \bigcup_k Z_k E_{kj} \cup B_j = Z_j$, deoarece în expresia $\cup (Z_2)_k . (E_2)_{kj} \cup \bigcup (Z_1)_j (E_1)_{kj}$ prima reuniune ne dă partea din reuniunea $\cup Z_k E_{kj}$ care corespunde indicilor k-neciclici, iar a 2-a, partea corespunzătoare indicilor k-ciclici.

Conform teoremei I.6.3 matricea E_2 fiind nesingulară, avem :

$$Z_2 = (Z_1 E_1 \cup B_2) E_2^* .$$

Prin urmare $Z_0 = Z_1 \cup Z_2 = Z_1 \cup (Z_1 E_1 \cup B_2) E_2^* = Z_1 (U \cup E_1 E_2^*) \cup B_2 E_2^*$.

Cum $E_1 \subseteq E$, $E_2 \subseteq E$, $B_2 \subseteq B$ avem

$$Z_0 \subseteq Z_1 E^* \cup BE^* = (Z_1 \cup B) E^* .$$

În baza primei relații (I.6.5) găsim :

$$Z_0 \subseteq (BE^* \cup T_1 \cup B) E^* = (B \cup T_1) E^* ,$$

și cum avem evident

$$Z_0 = (B \cup T) E^* \supseteq (B \cup T_1) E^*,$$

obținem

$$Z_0 = (B \cup T_1) E^*. \blacksquare$$

Consecința I.6.2. Dacă $e \in E$ atunci mulțimea soluțiilor ecuației $Z = ZE \cup B$ este mulțimea limbajelor de forma $Z = (B \cup T) E^*$ unde T este un limbaj arbitrar.

Observația I.6.5. Teorema I.6.3 este un caz particular al teoremei I.6.6 deoarece dacă matricea E este nesingulară nici un indice nu este ciclic cu privire la E.

§ I. 7. FUNCȚII RECURSIVE

Construcția clasei funcțiilor recursive pornește de la considerarea unor funcții numite *funcții primare*. Aceste din urmă funcții sînt:

1) *funcția de succesiune* $\mathbf{s} : N \to N^*$, dată de $\mathbf{s}(n) = n + 1$, $\forall n \in N$,

2) *funcția nulă* $\mathbf{o}^k : N^k \to \{0\}$, dată de $\mathbf{o}^k(x_1, \ldots, x_k) = 0$, $\forall (x_1, \ldots, x_k) \in N^k$, $k \in N$,

3) *funcția de proiecție* $\mathbf{i}_m^k : N^k \to N$, dată de $\mathbf{i}_m^k(x_1, \ldots, x_k) = x_m$, $\forall (x_1, \ldots, x_k) \in N^k$, $1 \leqslant m \leqslant k$, $k, m \in N$.

Vom nota $\mathbf{o}^1 = \mathbf{o}$.

Definiția I.7.1. *O funcție numerică parțială* sau pe scurt o *funcție parțială* este o aplicație $f : D \to N$, unde $D \subseteq N^k$. Dacă $D = N^k$, f este o *funcție total definită*.

Fie $f_1, \ldots, f_n : D \to D_1$, cu $D \subseteq N^k$, $D_1 \subseteq N$ și $g : D_1^n \to N$ un număr de $n + 1$ funcții parțiale.

Definiția I.7.2. *Operatorul de superpoziție* asupra a $n + 1$ *funcții parțiale*, $\mathbf{S}^{n+1}$ atașează funcțiilor $g, f_1, \ldots, f_n$ funcția $f : D \to N$ dată de:

$$f(x_1, \ldots, x_k) = g(f_1(x_1, \ldots, x_k), \ldots, f_n(x_1, \ldots, x_k)).$$

Vom nota $f = \mathbf{S}^{n+1}(g, f_1, \ldots, f_n)$.

Fie $F = \{f_1^{n_1}, \ldots, f_s^{n_s}\}$ o mulțime de funcții parțiale, funcția $f_j^{n_j}$ având n_j argumente, $j = \overline{1, s}$ și $P = \{\mathbf{i}_m^n \mid m, n \in N\}$, mulțimea funcțiilor de proiecție.

Definiția I.7.3. *Mulțimea funcțiilor F-elementare* este mulțimea funcțiilor ce se pot obține prin operatorul de superpoziție din funcțiile din $F \cup P$.

Fie funcțiile parțiale $g : D_g \to N$, $h : D_h \to N$ cu $D_g \subseteq N^n$, $D_h \subseteq N^{n+2}$.

Definiția I.7.4. *Operatorul de recursie* asupra funcțiilor g și h, $\mathbf{R}(g, h)$ atașează funcțiilor g și h funcția parțială $f : D_f \to N$, $D_f \subseteq N^{n+1}$ dată de

(I.7.1) $$f(x_1, \ldots, x_n, 0) = g(x_1, \ldots, x_n).$$

(I.7.2) $$f(x_1, \ldots, x_n, y + 1) = h(x_1, \ldots, x_n, y, f(x_1, \ldots, x_n, y)).$$

Observația I.7.1. În cazul $n = 0$ definiția funcției $f = \mathbf{R}(g, h)$ revine la :

$$f(0) = a.$$

$f(y + 1) = h(y, f(y))$, deoarece funcția g se reduce în acest caz la o constantă, notată cu a.

Observația I.7.2. Relațiile (I.7.1) și (I.7.2) definesc funcția f în mod unic. Avem astfel, pentru $b_k = f(x_1, \ldots, x_n, k)$ următorul mod de calcul.

$$b_0 = g(x_1, \ldots, x_n).$$

$$b_1 = h(x_1, \ldots, x_n, 0, b_0).$$

$$\cdot$$

$$\cdot$$

$$b_k = h(x_1, \ldots, x_n, k - 1, b_{k-1}).$$

$$\cdot$$

$$\cdot$$

Dacă $(x_1, \ldots, x_n, m) \notin D_f$, atunci $(x_1, \ldots, x_n, k) \notin D_f$ pentru $k \geq m$. Dacă funcțiile g și h sînt peste tot definite, atunci și f este peste tot definită.

Fie F o mulțime de funcții parțiale :

Definiția I.7.5. *Mulțimea funcțiilor F - primitiv recursive* este mulțimea funcțiilor care pot fi obținute din funcțiile aparținînd mulțimii F și din funcțiile primare $\{\mathbf{s}, \mathbf{o}, P\}$ printr-un număr finit de aplicări ale operatorilor de superpoziție și recursie primitivă.

Dacă $F = \varnothing$ funcțiile $\varnothing$ - primitiv recursive se numesc *funcții primitiv recursive.* Acestea se pot obține printr-un număr finit de aplicări ale operatorilor de superpoziție și recursie primitivă asupra funcțiilor primare $\{\mathbf{s}, \mathbf{o}, P\}$.

Observația I.7.3. Deoarece funcțiile primare sînt peste tot definite, funcțiile primitiv-recursive sînt de asemenea peste tot definite.

Observația I.7.4. Dacă $F_1 \subseteq F_2$ orice funcție F_1 - primitiv-recursivă este și F_2 - primitiv - recursivă. De aceea funcțiile primitiv-recursive sînt F - primitiv recursive oricare ar fi F.

Exemplul I.7.1. Conform definiției, funcțiile $\mathbf{s}$, $\mathbf{o}$, $\mathbf{i}_m^n$ sînt funcții primitiv-recursive. Deoarece

$$o^n = \mathbf{S}^2(\mathbf{o}, \mathbf{i}_1^n),$$

și funcția $\mathbf{o}^n$ este primitiv recursivă, $\forall n \in N^*$.

Exemplul I.7.2. Fie $k : N^n \to N$ o funcție constantă, $k(u_1, \ldots, u_n) = a$. Deoarece $k(u_1, \ldots, u_n) = \underbrace{\mathbf{s}(\mathbf{s}(\ldots \mathbf{s}}_{a}(o^n(u_1, \ldots, u_n))$, k este o funcție primitiv-recursivă.

Exemplul I.7.3. Funcția $f_s(x, y) = x + y$ este primitiv-recursivă. Într-adevăr, avem :

$$f_s(x, 0) = x = \mathbf{i}_1^1(x),$$

$$f_s(x, y + 1) = h(x, y, f(x, y)),$$

unde $h(x, y, z) = z + 1$. Deoarece $h(x, y, z) = \mathbf{s}(\mathbf{i}_3^3(x, y, z))$, h este o funcție primitiv-recursivă și $f = \mathbf{R}(\mathbf{i}_1^1, h)$.

Exemplul I.7.4. Funcția $f_p(x, y) = x\,y$ este primitiv-recursivă deoarece :

$$f_p(x, 0) = \mathbf{o}(x),$$

$$f_p(x, y + 1) = h_1(x, y, f_p(x, y)),$$

unde $h_1(x, y, z) = f_s(\mathbf{i}_1^3(x, y, z), \mathbf{i}_3^3(x, y, z))$.

Exemplul I.7.5. Funcția $f_e(x, y) = x^y$ este primitiv-recursivă. Astfel :

$$f_e(x, 0) = 1 = k(x),$$

$$f_e(x, y + 1) = h_2(x, y, f_e(x, y)),$$

unde
$h_2(x, y, z) = f_p(\mathbf{i}_1^3(x, y, z), \mathbf{i}_3^3(x, y, z))$, iar $k(x) = 1$ este o funcție constantă.

Exemplul I.7.6. Funcțiile :

$$\operatorname{sg} x = \begin{cases} 0, & \text{dacă } x = 0 \\ 1, & \text{dacă } x > 0 \end{cases} \qquad \text{și} \qquad \overline{\operatorname{sg}}\, x = \begin{cases} 1, & \text{dacă } x = 0 \\ 0, & \text{dacă } x > 0 \end{cases}$$

sînt funcții primitiv-recursive, deoarece, în baza observației I.7.1, avem

$$\operatorname{sg} 0 = 0,$$

$$\operatorname{sg}(x + 1) = h(x, \operatorname{sg} x).$$

unde $h(x, y) = 1$ este o funcție constantă. Apoi

$$\overline{\operatorname{sg}}\, 0 = 1,$$

$$\overline{\operatorname{sg}}\,(x + 1) = \bar{h}(x, \overline{\operatorname{sg}}\, x),$$

cu $\bar{h}(x, y) = 0$.

Exemplul I.7.7. Funcția „diferență pozitivă" $f_d(x, y) = x \dot{-} y$ dată de

$$x \dot{-} y = \begin{cases} x - y, & \text{dacă } x \geqslant y \\ 0, & \text{dacă } x < y \end{cases}$$

este o funcție primitiv-recursivă. Să arătăm mai întîi că funcția $\varphi(x) = x \dot{-} 1$ este o funcție primitiv-recursivă. Avem

$$\varphi(0) = 0,$$

$$\varphi(x + 1) = \chi(x, \varphi(x + 1)).$$

unde $\chi(x, y) = \mathbf{i}_1^3(x, y)$. Apoi

$$f_d(x, 0) = \mathbf{i}_1^1(x),$$

$$f_d(x, y + 1) = h(x, y, f_d(x, y)),$$

unde
$h(x, y, z) = \varphi(\mathbf{i}_1^3(x, y, z))$, aceasta deoarece $x \dot{-} (y \dot{+} 1) = (x \dot{-} y) \dot{-} 1$.

Ultima egalitate se justifică imediat. Obținem:

$$(x \dot{-} y) \dot{-} 1 = \begin{cases} (x - y) \dot{-} 1, & \text{dacă } x \geqslant y \\ 0, & \text{dacă } x < y \end{cases} = \begin{cases} x - y - 1, & \text{dacă } x \geqslant y + 1 \\ 0, & \text{dacă } x < y + 1 \end{cases} =$$

$$= x \dot{-} (y \dot{+} 1).$$

Exemplul I.7.8. Funcția $f_v(x, y) = |x - y|$ este primitiv-recursivă, deoarece $|x - y| = (x \dot{-} y) + (y \dot{-} x) = f_s(f_d(x, y), f_d(\mathbf{i}_2^2(x, y), \mathbf{i}_1^2(x, y)))$.

Funcția $f_m(x, y) = \min(x, y) = y \dot{-} (y \dot{-} x)$ este o funcție primitiv-recursivă, deoarece $f_m(x, y) = f_d(\mathbf{i}_2^2(x, y), f_d(\mathbf{i}_2^2(x, y), \mathbf{i}_1^2(x, y)))$.

Exemplul I.7.9. Funcția $f_r(x, y)$ dată de restul împărțirii lui y prin x este o funcție primitiv-recursivă. Astfel

$$f_r(x, 0) = \mathbf{o}(x),$$

$$f_r(x, y + 1) = h(x, y, f_r(x, y)),$$

unde $h(x, y, z) = f_p(\mathbf{s}(\mathbf{i}_3^3(x, y, z)), \operatorname{sg}(f_v(\mathbf{i}_1^3(x, y, z), \mathbf{s}(\mathbf{i}_3^3(x, y, z)))))$.

Exemplul I.7.10. Împărțirea întreagă $f_i(x, y)$, funcția care atașează perechii (x, y) partea întreagă a câtului împărțirii lui y la x este o funcție primitiv-recursivă, întrucât

$$f_i(x, 0) = \mathbf{o}(x),$$

$$f_i(x, y + 1) = h(x, y, f_i(x, y)),$$

unde $h(x, y, z) = f_s(\mathbf{i}_3^3(x, y, z), \overline{\operatorname{sg}}(f_v(\mathbf{i}_1^3(x, y, z), \mathbf{s}(f_r(\mathbf{i}_1^3(x, y, z), \mathbf{i}_2^3(x, y, z))))))$.

Observația I.7.5. Din definiția funcțiilor primitiv-recursive rezultă că o astfel de funcție este descrisă de un șir finit de simboli dintr-o mulțime numărabilă. Ca atare, mulțimea funcțiilor primitiv-recursive este numărabilă (v. teorema I.1.6).

Fie $f: D \to N$ cu $D \subset N^k$ o funcție parțială, pentru care există un procedeu de calcul al valorilor funcției. Convenim că $(x_1, \ldots, x_{k-1}, x_k) \notin D$ dacă și numai dacă procedeul considerat nu calculează într-un număr finit de pași valoarea $f(x_1 \ldots, x_k)$.

Fixăm primele $k - 1$ argumente și considerăm ecuația

$$f(x_1, \ldots, x_{k-1}, y) = x_k. \tag{I.7.3}$$

Pentru a găsi soluția ecuației (I.7.3), calculăm succesiv, aplicînd procedeul existent, valorile $f(x_1, \ldots, x_{k-1}, y)$ pentru $y = 0, 1, \ldots$

Să notăm cu:

(1.7.4) $\mu_y(f(x_1, \ldots, x_{k-1}, y) = x_k) = \min\{j \mid (f(x_1, \ldots, x_{k-1}, j) = x_k\}$.

Remarcăm că $\mu_y(f(x_1, \ldots, x_{k-1}, y) = x_k)$ nu-i definit atunci cînd:

1) $f(x_1, \ldots, x_{k-1}, 0)$ nu-i definit,

2) $f(x_1, \ldots, x_{k-1}, j)$ este definită și diferită de x_k pentru $j = \overline{0, m-1}$, iar $f(x_1, \ldots, x_{k-1}, m)$ nu-i definită,

3) $f(x_1, \ldots, x_{k-1}, j)$ este definită și diferită de x_k pentru orice $j \in N$.

În toate celelalte cazuri $\mu_y(f(x_1, \ldots, x_{k-1}, y) = x_k)$ este definit și este dat de cea mai mică soluție a ecuației

$$f(x_1, \ldots, x_{k-1}, j) = x_k.$$

Observația 1.7.6. Atragem atenția asupra faptului că egalitatea (1.7.4) are loc în mod condiționat, adică în cazul în care $\mu_y(f(x_1, \ldots, x_{k-1}, y) = x_k)$ este definit. Astfel $\mu_y(f(x_1, y) = x_2)$, unde $f(x_1, x_2) = x_2 \dot{-} (x_1 \dot{-} 1))$ nu-i definită, deoarece $f(x_1, 0)$ nu-i definită. Pe de altă parte avem:

$\min\{j \mid f(x_1, j) = 0\} = x_1 \dot{-} 1$, iar egalitatea (1.7.4) nu are loc.

Definiția 1.7.6. *Operatorul de minimizare* $\mathbf{M}$ atașează funcției f funcția parțială $\mathbf{M}(f): D_1 \to N$ dată de:

$\mathbf{M}(f)(x_1, \ldots, x_k) = \mu_y(f(x_1, \ldots, x_k) = \mu_y(f(x_1, \ldots, x_{k-1}, y) = x_k)$, unde $D_1 \subseteq N^k$.

Observația 1.7.7. Pentru $k = 1$, funcția parțială $\mathbf{M}(f)$ este funcția inversă f^{-1}.

Exemplul 1.7.11. Fie funcția $f_s(x_1, x_2) x + x_2$. Avem

$\mathbf{M}(f_s)(x_1, x_2) = \mu_y(x_1 + y = x_2) = x_2 - x_1$, unde $D_1 = \{(x_1, x_2) \mid x_2 \geqslant x_1\}$.

Exemplul 1.7.12. Funcția $\mathbf{M}(\mathrm{sg})$ este dată de

$$\mathbf{M}(\mathrm{sg})(x) = \mathrm{sg}^{-1}(x) = \begin{cases} x, \text{ dacă } x \leqslant 1, \\ \text{nedefinită, dacă } x > 1, \end{cases}$$

iar $\mathbf{M}(\mathbf{s})$ se obține din:

$$\mathbf{M}(\mathbf{s})(x) = \mathbf{s}^{-1}(x) = \begin{cases} x - 1, \text{ dacă } x > 0 \\ \text{nedefinită, dacă } x = 0. \end{cases}$$

Observația 1.7.8. Numerele:

$$\mu_y(|f(x_1, \ldots, x_k, y) - g(x_1, \ldots, x_k, y)| = 0,$$

$$\mu_y(\mathrm{sg}|f(x_1, \ldots, x_k, y) - g(x_1, \ldots, x_k, y)| = 1),$$

sînt de fapt respectiv numerele:

$$\mu_y(f(x_1, \ldots, x_k, y) = g(x_1, \ldots, x_k, y)),$$

$$\mu_y(f(x_1, \ldots, x_k, y) \neq g(x_1, \ldots, x_k, y)).$$

Fie $F = \{f_1, \ldots, f_n\}$ o familie de funcții parțiale.

Definiția I.7.7. O *funcție F-parțial recursivă* este o funcție parțială, care se obține din funcțiile din familia F și din funcțiile primare **s**, **o**, $\mathbf{i}_m^n$ printr-un număr finit de aplicări ale operatorilor de superpoziție, recursie primitivă și minimizare.

O funcție $\varnothing$-parțial recursivă se numește *funcție parțial-recursivă*.

Observația I.7.9. Orice funcție F-primitiv recursivă este și F-parțial recursivă. În particular, o funcție primitiv-recursivă este și parțial-recursivă.

Observația I.7.10. Clasa funcțiilor parțial-recursive include strict clasa funcțiilor primitiv-recursive, deoarece printre funcțiile parțial-recursive se află funcții parțial definite (v. exemplul I.7.10), în timp ce funcțiile primitiv-recursive sînt peste tot definite (v. observația I.7.3).

Fie $M \subseteq N$. Vom atașa mulțimii M două funcții numerice și anume funcția total definită:

$$h_M(n) = \begin{cases} 0, \text{ dacă } n \in M, \\ 1, \text{ dacă } n \notin M, \end{cases}$$

și funcția parțial definită:

$$h_M^p(M) = \begin{cases} 0, \text{ dacă } n \in M, \\ \text{nedefinită, în caz contrar.} \end{cases}$$

Observația I.7.11. Avem $h_M(n) = 1 - \chi_M(n)$. Apoi, dacă $M = \varnothing$, atunci $h_M(n) = 1$, $\forall n \in M$, iar $h_{\overline{M}}$ este peste tot nedefinită.

Definiția I.7.8. O mulțime $M \subseteq N$ este *primitiv-recursivă* (*parțial-recursivă*) dacă funcția h_M este primitiv-recursivă (dacă h_M^p este parțial-recursivă).

Observația I.7.12. Orice mulțime primitiv-recursivă este parțial-recursivă, deoarece $h_M^p(n) = 0 - h_M(n)$.

Observația I.7.13. Funcția numerică parțială:

$$f^p(n) = \begin{cases} f(n), \text{ dacă } n \in M, \\ \text{nedefinită dacă } n \notin M, \end{cases}$$

este parțial recursivă, unde f este o funcție primitiv-recursivă, iar $M \subseteq N$ o mulțime primitiv-recursivă. Într-adevăr, funcția h_M^p este o funcție parțial recursivă (v. observația I.7.12), iar $f^p(n) = f(n) + h_M^p(n)$.

Observația I.7.14. Următorul enunț este cunoscut sub numele de *teza lui Alonso Church*: „*Clasa funcțiilor numerice pentru care există algoritmi de calcul coincide cu clasa funcțiilor parțial-recursive*".

Teorema I.7.1. *Fie $f: D \to N$, $D \subseteq N^n$ o funcție primitiv (parțial) recursivă. Atunci funcția $g: D_1 \to N$, $D_1 \subseteq N^n$ definită prin:*

$$g(x_1, \ldots, x_n) = \sum_{j=0}^{x_n} f(x_1, \ldots, x_{n-1}, j) \tag{I.7.5}$$

este de asemenea primitiv-recursivă (parțial-recursivă).

Demonstrație. Din egalitatea (I.7.5) obținem :

$$g(x_1, ..., x_{n-1}, 0) = f(x_1, ..., x_{n-1}, 0),$$

$$g(x_1, ..., x_{n-1}, y+1) = g(x_1, ..., x_{n-1}, y) + f(x_1, ..., x_{n-1}, y+1)$$

și deci funcția g se obține prin recursie primitivă din funcțiile recursive $f(x_1, ..., x_{n-1}, 0)$, $h(x_1, ..., x_{n-1}, y, z) = z + f(x_1, ..., x_{n-1}, y+1)$ și de aceea g este primitiv (parțial) recursivă. ■

Teorema I.7.2. *Dacă funcția $f : N^n \to N$ este primitiv recursivă, atunci funcția $g : N^n \to N$ dată prin*

$$g(x_1, ..., x_n) = \prod_{j=0}^{x_n} f(x_1, ..., x_{n-1}, j), \tag{1.7.6}$$

este de asemenea primitiv-recursivă.

Demonstrație. Se procedează ca în teorema precedentă cu $h(x_1, ..., x_{n-1}, y, z) = zf(x_1, ..., x_{n-1}, y+1)$. ■

Teorema I.7.3. *Fie $g : N^{n+1} \to N$ și $\alpha : N^n \to N$ două funcții primitiv-recursive, astfel încît ecuația $g(x_1, ..., x_n, y) = 0$ să aibă cel puțin o soluție în y, oricare ar fi $(x_1, ..., x_n) \in N^n$ și*

$$\mu_y (g(x_1, ..., x_n, y) = 0) \leqslant \alpha(x_1, ..., x_n), \quad \forall (x_1, ..., x_n) \in N^n \tag{I.7.7.}$$

Atunci funcția $f : N^n \to N$ dată de $f(x_1, ..., x_n) = \mu_y (g(x_1, ..., x_n) = 0)$ este de asemenea primitiv-recursivă.

Demonstrație. Fie $(x_1, ..., x_n) \in N^n$ și $a = \mu_y (g(x_1, ..., x_n, y) = 0)$. Considerăm șirul $\pi_j = \prod_{k=0}^{j} g(x_1, ..., x_n, k)$, $j \in N$. Din definiția lui a rezultă că $\pi_0, \pi_1, ... \pi_{a-1}$, sînt nenuli și $\pi_{a+m} = 0$, $\forall m \in N$. Astfel, din condiția (I.7.7) obținem $a = \sum \{\operatorname{sg} \pi_j \mid j = \overline{0, \alpha(x_1, ..., x_n)}\}$. Introducînd funcția primitiv-recursivă $h(x_1, ..., x_n, z) = \prod_{j=0}^{z} g(x_1, ..., x_n, j)$ ultima egalitate se scrie sub forma :

$$f(x_1, ..., x_n) = \Sigma \{\operatorname{sg} h(x_1, ..., x_n, j) \mid j = \overline{0, \alpha(x_1, ..., x_n)}\}.$$

Afirmăm că f este o funcție primitiv-recursivă. Într-adevăr, $\operatorname{sg} h(x_1, ..., x_n, z)$ este o funcție primitiv-recursivă în baza exemplului I.7.6. Apoi, în baza teoremei I.7.1, funcția $t(x_1, ..., x_n, v) = \sum_{j=0}^{v} \operatorname{sg} h(x_1, ..., x_n, j)$ este primitiv-recursivă și $f(x_1, ..., x_n) = t(x_1, ..., x_n, \alpha(x_1, ..., x_n))$. ■

Consecința I.7.1. Funcția $[\sqrt{x}]$ este primitiv-recursivă. Într-adevăr, egalitatea $n = [\sqrt{x}]$ este echivalentă cu $n^2 \leqslant x < (n+1)^2$.
Astfel $[\sqrt{x}] = \mu_t (\operatorname{sg}((t+1)^2 \dot{-} x) = 1) < x$ și se folosește teorema precedentă.

Observația I.7.15. Am arătat (v. exemplul I.1.11) că mulțimea $N^* \times N^*$ este numărabilă. Cum $N \times N = (N^* \times N^*) \cup \{(0,0)\} \cup (\{0\} \times N^*) \cup (N^* \times \{0\})$, rezultă că și mulțimea $N \times N$ este numărabilă. Fie $N \times N = \{(0,0), (0,1), (1,0), (0,2), (1,1), (2,0), \ldots\}$ e (x, y) locul ocupat de perechea (x, y) în șirul de mai sus și $\mathbf{l}(n) = x$, $\mathbf{r}(n) = y$, dacă $\mathbf{e}(x, y) = n$. Avem $\mathbf{e}(x, y) = \frac{(x+y)(x+y+1)}{2} + x$ și $\mathbf{l}(n) = n - \frac{[\sqrt{8n+1}]+1}{2}\cdot\frac{[\sqrt{8n+1}]-1}{2}$ așa cum se poate constata folosind aceeași manieră de lucru ca în exemplul I.1.11. Similar, se obține un rezultat pentru funcția $\mathbf{r}(n)$ (exercițiu !).

Cum toate funcțiile care intră în expresiile lui $\mathbf{l}(n)$ și $\mathbf{r}(n)$ sînt funcții primitiv-recursive, conchidem că funcțiile $\mathbf{l}(n)$ și $\mathbf{r}(n)$ sînt funcții primitiv-recursive.

Cum mulțimea N^k este numărabilă, vom nota cu $\mathbf{e}^k(x_1, \ldots, x_k)$ locul k-uplului $(x_1, \ldots, x_k)$ într-o enumerare a acestei mulțimi obținută astfel : $\mathbf{e}^k(x_1, \ldots, x_k) = \mathbf{e}(\mathbf{e}(\ldots(\mathbf{e}(x_1, x_2), x_3)\ldots), x_k)$. Dacă $\mathbf{e}^k(x_1, \ldots x_k) = m$, atunci vom pune $\mathbf{e}_{kj}(m) = x_j$. Se observă că $x_k = \mathbf{r}(m)$, $x_{k-1} = \mathbf{r}(\mathbf{l}(m)), \ldots, x_2 = \mathbf{r}(\mathbf{l}(\ldots\mathbf{l}(m)))$ și, în fine, $x_1 = \mathbf{l}(\mathbf{l}\ldots(\mathbf{l}(m)))$. Astfel $\mathbf{e}$ și $\mathbf{e}_{kj}$ sînt funcții primitiv-recursive.

Ne-am ocupat pînă acum de funcții numerice, adică de funcții care aplică semigrupul $(N, +)$ în el însuși. Apare deci naturală extinderea acestei noțiuni la aplicații între semigrupuri libere oarecare.

Fie deci I^* semigrupul liber generat de alfabetul I.

Definiția I.7.9. *Funcțiile primare pe alfabetul* I sînt :

1) *funcțiile de succesiune* sînt funcțiile de forma $\mathbf{s}_i : I^* \to I^*$, date de $\mathbf{s}_i(p) = pi$, $\forall i \in I$, $p \in I^*$.

2) *funcția nulă* $\mathbf{o}^k : (I^*)^k \to I^*$ dată de $\mathbf{o}^k(p_1, \ldots, p_n) = e$, $\forall p_j \in I^*$, $j = \overline{1, n}$. Vom nota $\mathbf{o}^1 = \mathbf{o}$.

3) *funcția de proiecție* $\mathbf{i}_m^k : (I^*)^k \to I^*$, unde $\mathbf{i}_m^k(p_1, \ldots, p_k) = p_m$, $1 \leqslant m \leqslant k$, $\forall p_j \in I^*$, $j = \overline{1, m}$.

Analog cu operatorii $\mathbf{S}^{n+1}$, $\mathbf{R}$, $\mathbf{M}$ considerați anterior, vom introduce aici :

Definiția I.7.10. *Operatorul de superpoziție asupra* a $n+1$ *funcții parțiale*, $\mathbf{S}_{n+1}$, atașează funcțiilor $g, f_1, \ldots, f_n$ cu $g : D_g \to I$, $f_j : D_{f_j} \to I^*$ funcția $f : D \to I^*$ dată de

$$f(p_1, \ldots, p_k) = g(f_1(p_1, \ldots, p_k), \ldots, f_n(p_1, \ldots, p_k)), \text{ unde}$$

$$D_g \subseteq (I^*)^n, \quad D_{f_j} \subseteq (I^*)^k \quad \forall j = \overline{1, n}, \quad D \subseteq (I^*)^k.$$

Operatorul de i-recursie primitivă asupra funcțiilor g și h, $\mathbf{R}_i(g, h)$ atașează funcțiilor g și h funcția $f : D_f \to I^*$, $D_f \to (I^*)^{n+1}$ dată de :

$$f(p_1, \ldots, p_n, e) = g(p_1, \ldots, p_n),$$

$$f(p_1, \ldots, p_n, qi) = h(p_1, \ldots, p_n, i, f(p_1, \ldots, p_n, q)).$$

Acest operator se definește pentru orice $i \in I$.

Operatorul de i-minimizare $\mathbf{M}_i$ atașează funcției $f: D \to I^*$ funcția parțială $\mathbf{M}(f): D_1 \to I^*$ dată de:

$$\mathbf{M}(f)(p_1, ..., p_k) = \mu_i [f(p_1, ..., p_{k-1}, y) = e],$$

unde $y \in \{i\}^*$, iar $\mu_i \{f(p_1, ..., p_{k-1}, y) = e] = i^n$ cu $n = \min \{m \mid f(p_1, ..., p_{k-1}, \underbrace{i \ldots i}_{m} = e\}$,

iar $(p_1, ..., p_{k-1}, i^h) \in D_f$, $1 \leqslant h \leqslant k$.

Noțiunile pe care le prezentăm în continuare joacă un rol important în studiul unor limbaje (v. § II.12–14).

Definițiile I.7.5 și I.7.7 se extind corespunzător la funcții definite pe semigrupuri libere.

Să presupunem că $|I| = k$, $I = i_1, i_2, ..., i_k$. Cuvîntului $p = i_{j_1} i_{j_2} \ldots i_{j_m}$ i se atașează numărul natural $n = (j_1 - 1) + (j_2 - 1)k + \ldots + (j_m - 1)k^m$ scris în baza k. Corespondența astfel stabilită este o bijecție între semigrupul liber I^* și mulțimea numerelor naturale.

Vom putea scrie de aceea, în continuare, în locul cuvîntului p numărul natural n, care îi corespunde prin procedeul de mai sus.

Definiția I.7.11. O *mulțime recursivă* $E \subseteq I^*$ este o submulțime a lui I^*, a cărei funcție caracteristică:

$$\chi_E(n) = \begin{cases} 1, & \text{dacă } n \in E \\ 0, & \text{dacă } n \notin E, \end{cases}$$

este o funcție recursivă.

Definiția I.7.12. O *mulțime recursiv enumerabilă* $E \subseteq I^*$ este o submulțime a lui I^*, pentru care există o funcție recursivă, total definită, $f: I^* \to I^*$, astfel încît

$$E = \{f(n) \mid n \in I^*\}, \text{ sau } E = \varnothing.$$

Observația I.7.16. Deoarece I^* este o mulțime numărabilă, orice parte $E \subseteq I^*$ este o mulțime numărabilă. Dacă E este, în plus, recursiv enumerabilă, aceasta înseamnă că există o funcție recursivă ale cărei valori sînt cuvintele din E. Astfel, pentru un anume cuvînt $p \in I^*$ (căruia îi corespunde numărul n), în cazul în care $p \in E$, construind succesiv șirul $f(0)$, $f(1)$,... vom ajunge, după un număr finit de pași, să avem $f(m) = n$. Vom putea astfel rezolva problema apartenenței cuvîntului p la E. Deoarece în baza tezei lui Church există un algoritm pentru calculul funcției f, rezultă că o mulțime E este recursiv enumerabilă, dacă și numai dacă există un algoritm care permite alcătuirea unei „liste" a cuvintelor din E.

Teorema I.7.4. *O mulțime $E \subseteq I^*$ este recursiv enumerabilă, dacă și numai dacă există o funcție recursivă, total definită, de două variabile, $g: I^* \times I^* \to I^*$, astfel încît $\exists x \in I^*$, $g(a, x) = 0$, atunci și numai atunci cînd $a \in E$.*

Demonstrație. Fie $E \subseteq I$ o mulțime recursiv enumerabilă, $E \neq \emptyset$. Există atunci o funcție recursivă, total definită $f: I^* \to I^*$ cu $E = \{f(n) \mid n \in I^*\}$. Fie atunci funcția $h: I^* \times I^* \to I^*$ dată de:

$$h(p, q) = \begin{cases} 0, & \text{dacă } p = q \\ 1, & \text{dacă } p \neq q \end{cases} = \operatorname{sg}((p \dot{-} q) \quad (q \dot{-} p)).$$

Se constată imediat că h este o funcție recursivă, total definită. Funcția $g(p, q) = h(p, f(q))$ joacă rolul din enunț.

Invers, să presupunem că pentru mulțimea $E \subseteq I^*$ există o funcție g cu proprietatea cerută, adică E este mulțimea acelor cuvinte $a \in I^*$ pentru care ecuația $g(a, x) = 0$ este rezolvabilă.

Considerăm funcția primitiv recursivă $k(n) = \mathbf{l}(n) \overline{\operatorname{sg}}\, g(\mathbf{l}(n), \mathbf{r}(t)) + b \operatorname{sg} g(\mathbf{l}(t), \mathbf{r}(t))$, unde $\mathbf{l}(t)$ și $\mathbf{r}(t)$ sînt funcțiile introduse în observația I.7.15, iar $b \in E$. Afirmăm că $E = \{k(n) \mid n \in I^*\}$. Într-adevăr, dacă pentru un $n \in I^*$ avem $g(\mathbf{l}(n), r(n)) = 0$, atunci $\mathbf{l}(n) \in E$ și, conform definiției lui k, avem $k(t) = \mathbf{l}(t) \in E$, iar dacă $g(\mathbf{l}(n), \mathbf{r}(n)) \neq 0$, atunci $k(t) = b \in E$. Astfel $E \supseteq \{k(n) \mid n \in I^*\}$. Fie acum $a \in E$. Atunci există $x \in I^*$ cu $g(a, x) = 0$. Punînd $n = \mathbf{c}(a, x)$ și, deoarece $\mathbf{l}(\mathbf{c}(a, x)) = a$, obținem $k(n) = a$, deci $E \subseteq \{k(n) \mid n \in I^*\}$. ■

Consecința I.7.2. Fie $G(a, x_1, ..., x_n)$ o funcție primitiv-recursivă. Mulțimea acelor valori ale parametrului a pentru care ecuația $G(a, x_1, ..., x_n) = 0$ are cel puțin o soluție $(x_1, ..., x_n)$, este recursiv enumerabilă. Să introducem funcția primitiv recursivă $g(a, x) = G(a, \mathbf{c}_{n1}(x), ..., \mathbf{c}_{nn}(x))$, unde $\mathbf{c}_{ij}$ sînt funcțiile introduse în observația I.7.15. Deoarece $g(a, \mathbf{c}^n(x_1, ..., x_n)) = G(a, x_1, ..., x_n)$, urmează că mulțimea valorilor parametrului a, pentru care ecuația $g(a, x) = 0$ admite o soluție în x, coincide cu mulțimea acelor a pentru care ecuația $G(a, x_1, ..., x_n) = 0$ are soluții în $x_1, ..., x_n$.

Teorema I.7.5. *Reuniunile și intersecțiile finite de mulțimi enumerabile sînt mulțimi enumerabile.*

Demonstrație. Fie $E_1, ..., E_n \subseteq I^*$, n mulțimi recursiv enumerabile și $g_1(a, x), ..., g_n(a, x)$ funcțiile primitiv recursive atașate acestor mulțimi conform teoremei I.7.4. Atunci totalitatea valorilor parametrului a pentru care ecuațiile $g_1(a, x_1) ... g_n(a, x_n) = 0$ și $g_1(a, x_1) + ... + g_n(a, x_n) = 0$ coincid respectiv cu $A = \cup \{E_j \mid j = 1, n\}$ și $B = \cap \{E_j \mid j = 1, n\}$.
În baza consecinței I.7.2, mulțimile A și B sînt recursiv enumerabile.

Teorema I.7.6. *O mulțime recursivă este recursiv enumerabilă.*

Demonstrație. Fie E o mulțime recursivă. Atunci funcția ei caracteristică χ_E este o funcție primitiv recursivă și ecuația $1 - \chi_E(a) + x = 0$ are o soluție în x ($x = 0$) atunci și numai atunci cînd $a \in E$. ■

Reciproca acestei teoreme ne este adevărată, cum se va arăta în § II.14 (v. teorema II.14.6). Totuși, are loc:

Teorema I.7.7. *Dacă mulțimile $E \subseteq I^*$ și $E^c = I^* \setminus E$ sînt recursiv enumerabile, atunci E și E^c sînt recursive.*

Demonstrație. Fie g și h două funcții primitiv recursive cu $E = \{a \mid a \in I^*, \exists x \in I^*, g(a, x) = 0\}$, $E^c = \{a \mid a \in I^*, \exists x \in I^*, h(a, x) = 0\}$. Deoarece pentru orice $a \in I^*$ una din ecuațiile $g(a, x) = 0$ sau $h(a, x) = 0$ este satisfăcută funcția recursivă

$$f(a) = \mu_x (g(a, x)\, h(a, x) = 0)$$

este peste tot definită. Prin urmare, funcția $G(a) = 1 - \operatorname{sg} g(a, f(a))$ este de asemenea o funcție recursivă, peste tot definită. Dacă $G(a) = 1$, avem $\operatorname{sg} g(a, f(a)) = 0$ sau $g(a, f(a)) = 0$, de unde $a \in E$.

Invers, fie $a \in E$. Atunci ecuația $g(a, x) = 0$ are o soluție. Cum $a \notin E^c$ avem $h(a, x) \neq 0$, $\forall x \in I^*$ și deci $\mu_x (g(a, x) = 0) = \mu_x (g(a, x)\, h(a, x) = 0) = f(a)$. Astfel $g(a, f(a)) = 0$ și $G(a) = 1$.

Funcția $H(a) = \operatorname{sg} g(a, f(a))$ este funcția caracteristică pentru mulțimea E^c. ■

§ 1. 8. ELEMENTE DE LOGICĂ MATEMATICĂ

Ne vom ocupa mai întîi de construcția axiomatică a calculului propozițiilor, instrument care permite algebrizarea unei importante părți a activității de gîndire.

Definiția I.8.1. *Simbolii de bază ai calculului propozițional* sînt :

i) *simbolii variabilelor propoziționale* $\{v_n \mid n \in N\}$

ii) *conectivele logice* $\neg$ și $\wedge$ (care se citesc „*nu*" și „*și*")

iii) *semnele de punctuație* (și) (paranteza stîngă și dreaptă)

Vom nota cu I_{prop} mulțimea simbolilor de bază ai calculului propozițiilor și vom pune în evidență limbajul furmulelor calculului propozițional $F_{\text{prop}} \subseteq I^*_{\text{prop}}$

Definiția I.8.2. *Limbajul formulelor* este dat de următoarele reguli :

i) $\{v_n \mid n \in N\} \subseteq F_{\text{prop}}$

ii) Dacă $\alpha, \beta \in F_{\text{prop}}$ atunci $\neg\alpha, (\alpha \wedge \beta) \in F_{\text{prop}}$.

iii) Un cuvînt din I^*_{prop} este o formulă dacă și numai dacă se obține printr-un număr finit de aplicări ale regulilor de mai sus.

Observația 1.8.1. F_{prop} este cel mai mic limbaj care conține variabilele propoziționale și care, odată cu cuvintele α și β conține cuvintele $\neg\alpha$ și $(\alpha \wedge \beta)$.

Exemplul 1.8.1. Următoarele șiruri de simboluri sînt formule : $\neg\alpha$, $\neg(\alpha \wedge \neg\beta)$, $\neg(\neg(\alpha \wedge \beta) \wedge \neg(\neg(\alpha \wedge \beta) \wedge \neg\beta))$.

Observația 1.8.2. Vom introduce trei prescurtări pentru formule din F_{prop}. Anume, vom nota $(\alpha \vee \beta)$ în loc de $\neg(\neg\alpha \wedge \neg\beta)$, cu $(\alpha \to \beta)$ vom nota $\neg(\alpha \wedge \neg\beta)$ și, în fine în loc de $((\alpha \to \beta) \wedge (\beta \to \alpha))$ vom pune $(\alpha \longleftrightarrow \beta)$.

Din punctul de vedere al calculului propozițional ne interesează numai dacă o anume propoziție este adevărată sau falsă.

Definiţia I.8.3. *O realizare a calculului propoziţiilor* este o aplicaţie $f: \{v_n, n \in N\} \to B_2$, unde B_2 este algebra Boole cu două elemente, $B_2 = \{0, 1, \cup, \cap, ^-\}$.

Observaţia I.8.3. O realizare a calculului propoziţiilor poate fi extinsă la o aplicaţie $f: F_{prop} \to B_2$ punînd în mod recursiv :

$$f(\alpha \wedge \beta) = f(\alpha) \cap f(\beta) \tag{I.8.1}$$

$$f(\neg \alpha) = \overline{f(\alpha)}. \tag{I.8.2}$$

De aici rezultă că :

$$f(\alpha \vee \beta) = f(\neg(\neg \alpha \wedge \neg \beta)) = \overline{(\overline{f(\alpha)} \cap \overline{f(\beta)})} = f(\alpha) \cup f(\beta), \tag{I.8.3}$$

unde semnele „$\cap$" şi „$\cup$" se referă la operaţiile de intersecţie şi reuniune din algebra Boole.

Definiţia I.8.4. *Formula α satisface realizarea f dacă* $f(\alpha) = 1$. Vom nota aceasta prin $f \models \alpha$.

Formula α este o *tautologie* dacă satisface *orice* realizare f. Vom nota cu **Taut** mulţimea tautologiilor.

Observaţia I.8.4. Faptul că $f(\alpha) = 1$ semnifică în mod intuitiv că formula α este adevărată pentru realizarea f. În caz contrar α se interpretează ca falsă pentru această realizare. Astfel, de exemplu formula $\alpha \wedge \beta$ este adevărată pentru realizarea f dacă şi numai dacă atît α cît şi β sînt adevărate pentru această realizare (v. formulele (I.8.1)).

Observaţia I.8.5. Dacă formula α este construită cu ajutorul variabilelor $v_1, ..., v_n$ şi $f(v_j) = g(v_j)$, $j = \overline{1, n}$ atunci $f(\alpha) = g(\alpha)$. Astfel formula α determină funcţia booleană $f_\alpha: B_2^n \to B_2$, unde $f_\alpha(t_1, ..., t_n) = f(\alpha)$, f fiind realizarea care aplică v_j pe elementul $t_j \in B_2$.

Rezultă de aici că α este o tautologie dacă şi numai dacă $f_\alpha(t_1, ..., t_n) = 1$, $\forall (t_1, ..., t_n) \in B_2^n$, de unde urmează că este decidabil[1] dacă o anume formulă este o tautologie.

Definiţia I.8.5. *Axiomele calculului propoziţional* sînt următoarele formule :

i) $\alpha \to (\beta \to \alpha)$

ii) $[\alpha \to (\beta \to \gamma)] \to [(\alpha \to \beta) \to (\alpha \to \gamma)]$

iii) $(\neg \alpha \to \neg \beta) \to (\beta \to \alpha)$

iv$_a$) $(\alpha \wedge \beta) \to \alpha$

iv$_b$) $(\alpha \wedge \beta) \to \beta$

v) $[\gamma \to \alpha] \to [(\gamma \to \beta) \to (\gamma \to [\alpha \wedge \beta])]$

vi$_a$) $\alpha \to (\alpha \vee \beta)$

vi$_b$) $\beta \to (\alpha \vee \beta)$

vii) $[\alpha \to \gamma] \to [(\beta \to \gamma) \to ([\alpha \vee \beta]) \to \gamma)]$

[1] o problemă este *decidabilă* dacă există un algoritm pentru rezolvarea ei.

Observația I.8.6. Axiomele calculului propozițional sînt tautologii. De exemplu, fie f o realizare a calculului propozițiilor. Avem :

$$f(\alpha \to (\beta \to \alpha)) = \overline{f(\alpha) \cap \overline{f(\beta \to \alpha)}} = f(\alpha) \cap \overline{\overline{f(\beta) \cap \overline{f(\alpha)}}} =$$

$$= \overline{f(\alpha) \cap f(\beta) \cap \overline{f(\alpha)}} = \overline{0} = 1 \text{ etc.}$$

Definiția I.8.6. *Formula* β *este o consecință imediată a* formulei α *și* $\alpha \to \beta$, dacă α, $\alpha \to \beta \in F_{\text{prop}}$. Acest procedeu de a deduce consecințe imediate din formule poartă numele de regula *„modus ponens"*.

Definiția I.8.7. *O demonstrare a unei formule* α este un șir finit de formule $\alpha_1, ..., \alpha_n$ astfel încît $\alpha_n = \alpha$ și α_k este sau o axiomă sau există $j, h < k$ astfel încît α_k să fie o consecință imediată a formulelor α_j și α_h, $\forall k = \overline{1, n}$; numărul n este *lungimea demonstrației.*

Formula α este *inferabilă* dacă există o demonstrație a ei. Vom nota acest lucru prin $\vdash \alpha$ iar cu **Dem** vom nota mulțimea formulelor inferabile.

Teorema I.8.1. *Avem* $\vdash \alpha \to \alpha$, $\forall \alpha \in F_{\text{prop}}$.

Demonstrație. Fie axioma ii) pentru $\alpha = \gamma$ și $\beta = (\alpha \to \alpha)$:

$$[\alpha \to ([\alpha \to \alpha] \to \alpha)] \to [(\alpha \to [\alpha \to \alpha]) \to (\alpha \to \alpha)].$$

În baza axiomei i) putem scrie tautologia :

$$\alpha \to [(\alpha \to \alpha) \to \alpha],$$

unde am luat $\beta = (\alpha \to \alpha)$

Rezultă, în baza regulii modus ponens tautologia :

$$(\alpha \to [\alpha \to \alpha]) \to (\alpha \to \alpha).$$

Apoi, luînd în i) $\beta = \alpha$ găsim $\alpha \to [\alpha \to \alpha]$, de unde aplicînd din nou regula modus ponens avem

$$\alpha \to \alpha. \blacksquare$$

Lăsăm în seama cititorului demonstrarea faptului că următoarele formule :

$$(\alpha \to \beta) \to [(\gamma \to \alpha) \to (\gamma \to \beta)]$$

$$\alpha \to (\alpha \to \beta)$$

$$(\alpha \wedge \beta) \to (\beta \wedge \alpha)$$

$$\alpha \to [\beta \to (\alpha \to \beta)]$$

$$[(\alpha \wedge \gamma) \vee (\beta \wedge \gamma)] \to [(\alpha \vee \beta) \wedge \gamma]$$

$$(\alpha \to \delta) \to [(\beta \to (\gamma \to \alpha)) \to (\beta \to (\gamma \to \delta))]$$

sînt tautologii.

Teorema I.8.2. *Avem* **Dem** $\subseteq$ **Taut.**

Demonstrație. Vom proceda prin inducție asupra lungimii demonstrației. Pentru $n = 1$ faptul rezultă din observația I.8.3.

Să presupunem acum că α și $\alpha \to \beta$ sînt tautologii și să arătăm că și β este o tautologie (modus ponens).

Avem $f(\alpha) = f(\alpha \to \beta) = 1$, de unde

$$f(\beta) = [f(\alpha) \cap \overline{f(\alpha)}] \cup f(\beta) = [f(\alpha) \cup f(\beta)] \cap$$
$$\cap [\overline{f(\alpha)} \cap f(\beta)] = [1 \cup f(\beta)] \cap f(\alpha \to \beta) = 1 \cap 1 = 1. \blacksquare$$

Consecința I.8.1. Nu există formulă α astfel încît atît α cît și $\neg\alpha$ să fie inferabile.

Într-adevăr, $f(\alpha) = 1$ dacă și numai dacă $f(\neg\alpha) = 0$ și de aceea α și $\neg\alpha$ nu pot fi simultan tautologii. Cum **Dem** $\subseteq$ **Taut** rezultă că α și $\neg\alpha$ nu pot aparține simultan cu **Dem**.

Observația I.8.7. Relația „$-<$"pe F_{prop} dată prin: $\alpha -< \beta$ dacă și numai dacă $\vdash (\alpha \to \beta)$ este o relație de preordine (reflexivă și tranzitivă). Într-adevăr, reflexitatea ei rezultă din teorema I.9.1, iar tranzitivitatea se obține remarcînd că din $\vdash (\alpha \to \beta)$ rezultă $f(\alpha) \leqslant f(\beta)$, oricare ar fi realizarea f. În schimb, această relație nu este antisimetrică, deoarece, dacă $\alpha, b \in F_{prop}$ sînt două formule distincte putem arăta că $\alpha \wedge \beta -< \beta \wedge \alpha$ și $\beta \wedge \alpha -< \alpha \wedge \beta$ deși $\alpha \wedge \beta$ și $\beta \wedge \alpha$ sînt în mod evident formule distincte.

Observația I.8.8. Relația $\equiv \subseteq F_{prop} \times F_{prop}$ dată prin $\alpha \equiv \beta$ dacă și numai dacă $\vdash (\alpha \to \beta)$ și $\vdash (\beta \to \alpha)$ este o echivalență. Vom nota cu $[\alpha]$ clasa de echivalență a formulei α. Relația „$\equiv$" este dată de „$\equiv$" $= \bigcap (\ker f \mid f \in B_2^{F_{prop}})$.

Observația I.8.9. Relația $\leqslant \subseteq F_{prop}/\equiv \times F_{prop}/\equiv$ dată de $[\alpha] \leqslant [\beta]$ dacă și numai dacă $\vdash (\alpha \to \beta)$ este o relație de ordine parțială pe $F_{prop}/\equiv$. Corectitudinea acestei definiri rezultă din faptul că dacă $\vdash (\alpha \to \beta)$, $\alpha \equiv \alpha'$, $\beta \equiv \beta'$ avem $\vdash (\alpha' \to \beta')$.

Avem $[\alpha] \leqslant [\beta]$ dacă și numai dacă pentru orice realizare f găsim $f(\alpha) \leqslant f(\beta)$.

Teorema I.8.3. *Mulțimea parțial ordonată* $(F_{prop}/\equiv, \leqslant)$ *este o algebră Boole și* $[\alpha] = 1$ *dacă și numai dacă* $\vdash \alpha$, *iar* $[\beta] = 0$ *dacă și numai dacă* $\vdash \neg\beta$.

Demonstrație. Fie $\alpha, \beta \in F_{prop}$. Să arătăm că $\sup\{[\alpha], [\beta]\} = [\alpha \vee \beta]$, $\inf\{[\alpha], [\beta]\} = [\alpha \wedge \beta]$.

În baza axiomei vi_a), avem $[\alpha] \leqslant [\alpha \vee \beta]$, $[\beta] \leqslant [\alpha \vee \beta]$. Dacă $[\alpha] \leqslant [\gamma]$, $[\beta] \leqslant [\gamma]$ avem $f(\alpha) \leqslant f(\gamma)$, $f(\beta) \leqslant f(\gamma)$ și, conform cu I.9.3 obținem $f(\alpha \vee \beta) \leqslant f(\gamma)$, $\forall f \in B_2^{F_{prop}}$ ceea ce implică $[\alpha \vee \beta] \leqslant [\gamma]$. Similar se procedează pentru infimum.

Distributivitatea rezultă din tautologiile:

$$\vdash [(\alpha \wedge \beta) \vee (\gamma \wedge \beta)] \to [(\alpha \vee \gamma) \wedge \beta],$$
$$\vdash [(\alpha \vee \beta) \wedge \beta] \to [(\alpha \wedge \beta) \vee (\gamma \wedge \beta)],$$

valabile pentru orice $\alpha, \beta, \gamma \in F_{prop}$ (exercițiu !)

Dacă $\vdash \alpha$ atunci $f(\alpha) = 1$ și pentru orice $\delta \in F_{prop}$ avem $f(\delta) \leqslant f(\alpha)$, de unde $[\alpha]$ este cel mai mare element al laticii $F_{prop} / \equiv$.

Similar se arată că dacă $\vdash \neg \beta$ atunci $[\beta]$ este cel mai mic element al laticii $F_{prop} / \equiv$.

În sfîrșit, deoarece

$$\vdash \alpha \vee \neg\alpha \quad \text{și} \quad \vdash \neg(\alpha \wedge \neg\alpha)$$

(v. observația I.9.2 și teorema I.9.1) găsim $[\alpha] \vee [\neg\alpha] = 1$ și $[\alpha] \wedge [\neg\alpha] = 0$, ceea ce indică faptul că $F_{prop} / \equiv$ este o algebră Boole. ■

Această algebră este cunoscută în literatură (v. Bell, J. L.; Slomson, A. B. [1]) sub denumirea de *algebră Lindenbaum.*

Următorul anunț este cunoscut sub numele de *teorema de completitudine a calculului propozițional.*

Teorema I.8.4. *Avem* **Dem = Taut.**

Demonstrație. Se va consulta [Bell, J. L.; Slomson A. B. [1], p. 43].

Definiția I.8.8. Alfabetul simbolilor de bază ai calculului predicatelor I_{pred} este mulțimea alcătuită din:

i) *simbolii variabilelor individuale* $\{v_n \mid n \in N\}$.
ii) *simbolii predicativi* $\{P_n \mid n \in N\}$
iii) *simbolul egalității* $=$
iv) conectivele logice $\neg$ și $\wedge$
v) cuantificatorul existențial $\exists$
vi) simbolii de punctuație ( ).

Vom considera *aplicația de graduare* $g : \{P_n, n \in N\} \to N$, iar $g(P_n)$ va fi numit gradul simbolului predicativ P_n. Pentru simplitate $g(P_n)$ va fi notat cu $g(n)$.

Simbolii logici sînt $\{\neg, \wedge, \exists\}$.

Definiția I.8.8. O *formulă atomică* este un șir de simboli de forma $x = y$ sau de forma $P_n(x_1, ..., x_{g(n)})$.

Definiția I.8.9. *Limbajul L al formulelor calculului predicatelor* este dat de următoarele reguli:

i) Formulele atomice fac parte din L
ii) Dacă $\varphi, \psi \in L$ și v este o variabilă atunci $\neg\varphi$, $(\varphi \wedge \psi)$, $(\exists v)\varphi \in L$
iii) Un șir de simboli de bază este o formulă dacă și numai dacă se obține printr-un număr finit de aplicări ale regulilor de mai sus.

Observația I.8.10. Limbajul L al formulelor calculului predicatelor este cea mai mică mulțime de cuvinte din semigrupul liber I^*_{pred} care conține toate formulele atomice și care, împreună cu φ și ψ conține pe $\neg\varphi$, $(\varphi \wedge \psi)$ și $(\exists v)\varphi$, oricare ar fi simbolul de variabilă v.

Observația I.8.11. Vom folosi și în calculul predicatelor simbolii $\wedge, \rightarrow, \longleftrightarrow$ cu aceleași semnificații ca și în calculul propozițiilor (v. observația I.8.2). În plus formula $\neg(\exists x)\neg\varphi$ va fi notată cu $(\forall x)\varphi$.

Este clar că dacă $\varphi, \psi \in L$ atunci $(\varphi \vee \psi)$, $(\varphi \rightarrow \psi)$, $(\varphi \longleftrightarrow \psi)$ și $(\forall x)\varphi$ sînt formule.

Definiţia I.8.10. *Un v-cuantificator* este un şir de forma $(\exists v)$. Domeniul unui v-cuantificator se defineşte recursiv astfel:

i) Domeniul lui $(\exists v)$ în formula $(\exists v)\varphi$ este $(\exists v)\ \varphi$.

ii) Domeniul lui $(\exists v)$ în $\neg\varphi\,(\exists v)\psi$ este acelaşi ca în $\varphi\,(\exists v)\psi$.

iii) Domeniul lui $(\exists v)$ în $\varphi\,(\exists v)\ \psi\wedge\chi$ sau $\chi\wedge\varphi\,(\exists v)\psi$ este acelaşi ca şi în $\varphi\,(\exists v)\ \psi$.

iv) Domeniul lui $(\exists v)$ în $(\exists y)\,\varphi\,(\exists v)\psi$ este acelaşi ca şi în formula $\varphi(\exists v)\psi$.

Exemplul I.8.2. Să considerăm formula $\neg\,(\exists v_0)\,[(\exists v_1)\,[(\exists v_2)\,P\,(v_1, v_2).\ P\,(v_1, v_0)]\vee\neg\,P\,(v_0, v_0)]$. Domeniul cuantificatorului $(\exists v_1)$ este $(\exists v_1)\,[(\exists v_2)\,P\,(v_1, v_2)\wedge P\,(v_1, v_0)]$.

Definiţia I.8.11. O *apariţie* a unei variabile v într-o formulă este o *apariţie legată* dacă ea apare în domeniul unui v-cuantificator. În caz contrar apariţia lui v este *liberă*. În aceeaşi formulă o variabilă poate apare şi liber şi legat.

Dacă în formula φ variabilele $x_1, \ldots, x_n$ sînt libere vom nota acest lucru cu $\varphi\,(x_1, \ldots, x_n)$.

O formulă *deschisă* este o formulă care nu conţine cuantificatori.

Închiderea formulei $\varphi\,(x_1, \ldots, x_n)$ este formula $(\forall x_1)\ldots(\forall x_n)\,\varphi\,(x_1, \ldots, x_n)$.

Definiţia I.8.12. O *propoziţie* este o formulă în care nu avem apariţii libere ale variabilelor.

Definiţia I.8.13. *Axiomele calculului predicatelor* sînt următoarele formule:

A. i) $\varphi\to(\psi\to\varphi)$

ii) $[\varphi\to(\psi\to\chi)]\to([\varphi\to\psi)\to(\varphi\to\chi)]$

iii) $(\neg\varphi\to\neg\psi)\to(\psi\to\varphi)$

iv$_a$) $(\varphi\wedge\psi)\to\varphi$

iv$_n$) $(\varphi\wedge\psi)\to\psi$

v) $(\chi\to\varphi)\to[(\chi\to\psi)\to(\chi\to[\varphi\wedge\psi])]$

vi$_a$) $\varphi\to(\varphi\vee\psi)$

vi$_b$) $\psi\to(\varphi\vee\psi)$

vii) $(\varphi\to\chi)\to[(\psi\to\chi)\to([\varphi\vee\psi]\to\chi)]$

B. i) $(\forall x)\ \varphi\,(x)\to\varphi\,(y)$

ii) $(\forall x)\,(\varphi\to\psi)\to(\varphi\to(\forall x)\,\psi)$

C. i) $(\forall x)\,(x = x)$

ii) $(\forall x)\,(\forall y)\,(x = y\to[\varphi\to\varphi'])$, unde φ este o formulă din L, iar φ' este rezultatul înlocuirii unora dintre apariţiile libere ale lui x cu y.

Definiţia I.8.14. *Procedeele de deducere ale calculului predicatelor* sînt:

i) *modus ponens*: formula ψ este o consecinţă imediată a formulelor φ şi $\varphi\to\psi$, oricare ar fi formulele φ şi ψ.

ii) *generalizarea* : formula $(\forall x)\varphi$ este o consecință imediată a formulei φ, unde φ este o formulă și x este o variabilă.

Definiția I.8.15. O *demonstrație* a unei formule φ este un șir fiind de formule $\varphi_1, \ldots, \varphi_n = \varphi$ astfel încît pentru orice formulă φ_i să avem una din următoarele trei situații :

i) φ_i este o axiomă.

ii) există $j, k < i$ astfel încît φ_i este o consecință imediată a formulelor φ_j și φ_k (prin regula modus ponens).

iii) există $j < i$ astfel încît φ_i este o consecință a formulei φ_j (prin regula de generalizare).

O *formulă* φ este *inferabilă* dacă există o demonstrație a ei. Acest lucru va fi notat prin $\vdash \varphi$, iar cu **DEM** vom nota mulțimea formulelor inferabile ale calculului predicatelor.

Teorema I.8.5. *Dacă φ este o formulă a calculului predicatelor care se obține dintr-o tautologie α a calculului propozițiilor prin înlocuirea variabilelor prin formule atunci* $\varphi \in$ **DEM**.

Demonstrație. Afirmația se obține observînd că axiomele A.i) – vii) sînt de fapt axiomele calculului propozițiilor în care simbolii variabilelor au fost înlocuiți cu formule, iar modus ponens este un procedeu de demonstrare comun calculului propozițiilor și calculului predicatelor. ■

Teorema I.8.6. *Pentru orice formulă $\varphi(x, y)$ a calculului predicatelor avem*

$$\vdash (\forall x)(\forall y)\varphi(x, y) \to (\forall y)(\forall x)\varphi(x, y).$$

Demonstrație. În baza axiomei B i) avem :

$$\vdash (\forall x)(\forall y)\varphi(x, y) \to (\forall y)\varphi(x, y)$$

și

$$\vdash (\forall y)\varphi(x, y) \to \varphi(x, y).$$

Cum formula $(\alpha \to \beta) \to [(\beta \to \gamma) \to (\alpha \to \gamma)]$ este o tautologie aplicînd teorema I.8.5 găsim :

$$\vdash [(\forall x)(\forall y)\varphi(x, y) \to (\forall y)\varphi(x, y)] \to [((\forall y)\varphi(x, y) \to \varphi(x, y)) \to$$
$$\to ((\forall x)(\forall y)\varphi(x, y) \to \varphi(x, y))].$$

În baza regulii modus ponens avem

$$\vdash ((\forall y)\varphi(x, y) \to \varphi(x, y)) \to ((\forall x)(\forall y)\varphi(x, y) \to \varphi(x, y)),$$

și aplicînd încă o dată aceeași regulă obținem

$$\vdash (\forall x)(\forall y)\varphi(x, y) \to \varphi(x, y).$$

Prin generalizare putem scrie :

$$\vdash (\forall x)(\forall y)\varphi(x, y) \to \varphi(x, y)$$

și astfel, prin axioma B ii) și prin regula modus ponens :

$$\vdash (\forall x)(\forall y)\varphi(x, y) \rightarrow (\forall x)\varphi(x, y)$$

În sfîrșit, folosind regula de generalizare, axioma B ii) și modus ponens avem

$$\vdash (\forall x)(\forall y)\varphi(x, y) \rightarrow (\forall y)(\forall x)\varphi(x, y). \blacksquare$$

Consecința I.8.2. Ordinea în care se plasează cuantificatorii atunci cînd se consideră închiderea formulei $\varphi(x_1, ..., x_n)$ este indiferentă.

Observația I.8.12. O formulă este inferabilă dacă și numai dacă închiderea ei este inferabilă. Acest fapt rezultă din axioma B i) și din existența regulii de generalizare.

Observația I.8.13. În calculul propozițiilor formulele aveau una din „valorile de adevăr" : 0 sau 1. În calculul predicatelor pentru a interpreta o formulă ca adevărată sau falsă trebuie să precizăm semnificația simbolilor predicativi precum și valorile atribuite simbolilor variabilelor.

Definiția I.8.16. O *structură relațională* este un cuplu $\mathfrak{A} = <A, \{\rho_n \mid n \in N\}>$ unde A, domeniul lui $\mathfrak{A}$ este o mulțime nevidă și ρ_n este o relație pe A oricare ar fi $n \in N$.

Structura relațională $\mathfrak{A}$ este o *realizare* a limbajului calculului predicatelor dacă $a(\rho_n) = g(n)$, unde $a(\rho_n)$ este aritatea relației ρ_n, $\forall n \in N$. Vom numi ρ_n *valoarea predicatului* P_n în realizarea $\mathfrak{A}$.

Definiția I.8.17. O $\mathfrak{A}$ - *valuare a variabilelor calculului predicatelor* este un șir de elemente din A :

$$x = (x_0, x_1, ..., x_n, ...)$$

Definiția I.8.18. Valuarea x satisface φ în $\mathfrak{A}$, ceea ce se notează cu $\mathfrak{A} \underset{x}{\models} \varphi$, dacă :

i) $\mathfrak{A} \underset{x}{\models} V_m = V_n$ dacă și numai dacă x_m este același cu x_n.

ii) $\mathfrak{A} \underset{x}{\models} P_n(v_{i_1}, ..., v_{i_{\delta(n)}})$ dacă și numai dacă $< x_{i_1}, ..., x_{i_{\delta(n)}} > \in \rho_n$.

iii) $\mathfrak{A} \underset{x}{\models} \neg\varphi$ dacă și numai dacă nu avem $\mathfrak{A} \underset{x}{\models} \varphi$

iv) $\mathfrak{A} \underset{x}{\models} \varphi \wedge \psi$ dacă și numai dacă $\mathfrak{A} \underset{x}{\models} \varphi$ și $\mathfrak{A} \underset{x}{\models} \psi$

v) $\mathfrak{A} \underset{x}{\models} (\exists v_n)\varphi$ dacă și numai dacă există un a astfel încît $\mathfrak{A} \underset{x_n(a)}{\models} \varphi$, unde $x_n(a) = (x_1, ..., x_{n-1}, a, x_{n+1}, ...)$

Vom scrie $\mathfrak{A} \models \varphi[x_0, ..., x_n]$ dacă orice valuare care atribuie simbolului v_i valoarea x_i, $i \leqslant n$ satisface pe φ în $\mathfrak{A}$. Formula φ este *valabilă* în $\mathfrak{A}$ dacă $\mathfrak{A} \models \varphi$ oricare ar fi x, ceea ce se va nota cu $\mathfrak{A} \models \varphi$. Dacă $\mathfrak{A} \models \varphi$ oricare ar fi $\mathfrak{A}$ atunci φ este o formulă *universal valabilă*.

Se poate arăta :

Teorema I.8.7. *Mulțimea formulelor universal valabile coincide cu* **DEM**.

Demonstrație. Vezi [Bell, J. L. ; Slomson, A. B. 1, p. 63]. ■

PARTEA A II-A

LIMBAJE FORMALE

§ II. 0. Introducere

Limbajele formale constituie modele matematice atît ale limbajelor naturale, cît și ale limbajelor artificiale, folosite pentru comunicarea cu mașinile electronice de calcul (limbaje de programare).

Se prezintă o ierarhizare a claselor de limbaje formale unanim cunoscută și anume ierarhia lui Chomsky. Fiecare clasă a acestei ierarhii este studiată, atît din punct de vedere al generării limbajelor care fac parte din ea, cît și din punct de vedere al recunoașterii acestor limbaje cu ajutorul unor „mașini matematice". Sînt cercetate de asemenea proprietățile de închidere ale claselor de limbaje față de diferite operații, precum și raportul dintre diversele clase.

§ II. 1. GRAMATICI

Noțiunea matematică de gramatică este o formalizare a metodelor de construire a limbajelor asupra unor alfabete.

Definiția II.1.1. O *gramatică* este un 4-uplu $G = (I_N, I_T, x_0; F)$ unde I_N și I_T sînt două mulțimi disjuncte, finite numite respectiv *alfabet neterminal* și *alfabet terminal*, $x_0 \in I_N$ este *simbolul inițial*, iar F este *mulțimea regulilor de generare* adică o mulțime finită de perechi ordonate (p, q) cu $p \in (I_N \cup I_T)^* \setminus \{e\}$, p conținînd cel puțin un simbol din I_N, iar $q \in (I_N \cup I_T)^*$.

Dacă $(p, q) \in F$ vom nota acest lucru cu $p \to q$.

Definiția II.1.2. *Cuvîntul p generează direct cuvîntul q în gramatica G,* dacă $p = r_1 p' r_2$, $q = r_1 q' r_2$, $p' \to q'$ și $r_1, r_2 \in I^*$, unde $I = I_N \cup I_T$, ceea ce se va nota cu $p \underset{G}{\Rightarrow} q$.

Cuvîntul p generează cuvîntul q în gramatica G dacă există un şir finit de cuvinte $p = p_0, p_1, ..., p_n = q$, $p_j \in I^*$, $j = \overline{0, n}$ astfel ca $p_j \underset{G}{\Rightarrow} p_{j+1}$, $j = \overline{0, n-1}$.

Vom nota aceasta cu $p \overset{\bullet}{\underset{G}{\Rightarrow}} q$.

Şirul $p_0, p_1, ..., p_n$ formează o *deducţie* a cuvîntului p_n din cuvîntul p_0 în gramatica G, iar n este lungimea deducţiei.

Vom mai folosi şi notaţia:

$$p_0 \underset{G}{\Rightarrow} p_1 \underset{G}{\Rightarrow} \dots \underset{G}{\Rightarrow} p_n.$$

Observaţia II.1.1. Relaţia „$\overset{\bullet}{\underset{G}{\Rightarrow}}$" este închiderea tranzitivă şi reflexivă a relaţiei „$\underset{G}{\Rightarrow}$". În cazul în care nu există pericol de confuzie vom nota, pentru simplitate, relaţiile de mai sus prin $\overset{\bullet}{\Rightarrow}$ respectiv $\Rightarrow$.

Definiţia II.1.3. *Limbajul generat de gramatica G* este mulţimea:

$$L(G) = \{p \mid p \in I_T^*,\ x_0 \overset{\bullet}{\Rightarrow} p\},$$

adică $L(G)$ constă din mulţimea cuvintelor care sînt generate în gramatica G din simbolul iniţial x_0.

Definiţia II.1.4. Două *gramatici* G_1 şi G_2 sînt *echivalente* dacă $L(G_1) = L(G_2)$.

Exemplul II.1.1. Fie gramatica $G_1 = (\{x_0\}, \{i_1, i_2\}, x_0; \{x_0 \to i_1 i_2, x_0 \to i_1 x_0 i_2\})$. Atunci $L(G_1) = \{i_1^n i_2^n \mid n \in N^*\}$.

Într-adevăr, $i_1 i_2 \in L(G_1)$ deoarece avem $x_0 \Rightarrow i_1 i_2$. Apoi $i_1^2 i_2^2 = i_1 i_1 i_2 i_2 \in L(G_1)$ din deducţia $x_0 \Rightarrow i_1 x_0 i_2 \Rightarrow i_1 i_1 i_2 i_2$. Avem $i_1^3 i_2^3 \in L(G_1)$ deoarece $x_0 \Rightarrow i_1 x_0 i_2 \Rightarrow i_1 i_1 x_0 i_2 i_2 \Rightarrow i_1 i_1 i_1 i_2 i_2 i_2$, ş. a. m. d.

Exemplul II.1.2. Gramatica $G_2 = (\{x_0\}, I_T, x_0; \{x_0 \to p_1, ..., x_0 \to p_n\})$, unde $p_j \in I_T^*$, $j = \overline{1, n}$, generează limbajul finit $L(G_2) = \{p_1, ..., p_n\}$.

Exemplul II.1.3. Gramatica $G_{(n)} = (\{x_0\}, \{i_1, ..., i_n, k_1, ..., k_n\}, x_0; x_0 \to e, x_0 \to x_0 i_1 x_0 k_1 x_0, x_0 \to x_0 i_n x_0 k_n x_0\})$, $n \in N^*$, generează *limbajul n-paranteticesau n-limbajul de tip Dyck.*

Luînd $n = 1$ avem $G_{(1)} = (\{x_0\}; \{i_1, k_1\}, x_0; \{x_0 \to e, x_0 \to x_0 i_1 x_0 k_1 x_0\})$. Iată cîteva cuvinte din $L(G_{(1)})$ cu deducţiile lor:

$$x_0 \Rightarrow x_0 i_1 x_0 k_1 x_0 \Rightarrow x_0 i_1 k_1 x_0 \Rightarrow i_1 k_1 x_0 \Rightarrow i_1 k_1,$$

$$x_0 \Rightarrow x_0 i_1 x_0 k_1 x_0 \Rightarrow x_0 i_1 x_0 k_1 x_0 i_1 x_0 k_1 x_0 \overset{\bullet}{\Rightarrow} i_1 k_1 i_1 k_1,$$

$$x_0 \Rightarrow x_0 i_1 x_0 k_1 x_0 \Rightarrow x_0 i_1 x_0 i_1 x_0 k_1 x_0 k_1 x_0 \overset{\bullet}{\Rightarrow} i_1 i_1 k_1 k_1.$$

În n-limbajele de tip Dyck în locul simbolurilor $i_1 \ldots i_n$, $k_1, \ldots, k_n$ se pot folosi paranteze de diferite tipuri $(_1, \ldots, (_n,)_1, \ldots,)_n$ de unde denumirea de limbaje n-parantetice pentru aceste limbaje.

De exemplu, pentru 1-limbaje se folosesc în locul lui i_1, k_1 respectiv parantezele „(" și „)". Corespunzător, cuvintele considerate mai sus se scriu ( ), ( ) ( ) și (( )).

Exemplul II.1.4. Fie gramatica $G_3 = (\{x_0\}, \{v, 1, \wedge, \vee, \neg\}, x_0; \{x_0 \to x_0\ 1,\ x_0 \to v\ 1,\ x_0 \to \wedge\ x_0\ x_0,\ x_0 \to \vee\ x_0\ x_0,\ x_0 \to \neg\ x_0\})$.

Limbajul $L(G_3)$ constă din formulele bine construite ale calculului propozițional.

Astfel, primele două reguli de generare ne permit construirea variabilelor „indexate" $v1$, $v11$, $v111$,... Regulile de generare $x_0 \to \wedge\ x_0\ x_0$ și $x_0 \to \vee\ x_0\ x_0$ permit construirea conjuncției respectiv a disjuncției a două variabile, iar $x_0 \to \neg\ x_0$ construirea negației unei variabile oarecare.

Exemplul II.1.5. Fie gramatica $G_4 = (\{x_0\}, \{i_1, i_2, \ldots, i_n, \vee, *, \varnothing, (,)\}, x_0; \{x_0 \to i_1, \ldots, x_0 \to i_n, x_0 \to \varnothing, x_0 \to x_0 \vee x_0, x_0 \to (x_0\ x_0), x_0 \to x_0^*\})$. Limbajul $L(G_4)$ constă din expresiile regulate pe alfabetul $\{i_1, \ldots, i_n\}$ (v. I. Creangă, C. Reischer, D. Simovici [1], §II.14).

§ II.2. IERARHIA GRAMATICILOR

Gramaticile pot fi clasificate impunînd anumite restricții regulilor de generare.

Definiția II.2.1. *Gramaticile de tip 0* sînt gramaticile care nu au restricții asupra regulilor de generare.

Gramaticile de tip 1 sînt gramaticile ale căror reguli de generare au forma :

$$p_1\, x\, p_2 \to p_1\, p p_2,$$

unde

$$p_1, p_2 \in (I_N \cup I_T)^*,\ x \in I_N \text{ și } p \in (I_N \cup I_T)^* \setminus \{e\},$$

sau

$$x \to e,$$

și $x \in I_N$ nu apare în partea dreaptă a vreunei reguli de generare.

Gramaticile de tip 2 sînt gramaticile ale căror reguli de generare sînt de forma :

$$x \to p,$$

unde

$$x \in I_N,\ p \in (I_N \cup I_T)^*.$$

Gramaticile de tip 3 sînt gramaticile ale căror reguli de generare sînt de forma :

$$x \to px_1 ,$$

unde

$$x, x_1 \in I_N \text{ și } p \in I_T^* ,$$

sau

$$x \to p,$$

unde $x \in I_N$, $p \in I_T^*$.

Definiția II.2.2. Un *limbaj de tip j* este un limbaj generat de o gramatică de tip j, $j = \overline{0,3}$

Vom nota cu $\mathcal{L}_j$ mulțimea limbajelor de tip j.

Observația II.2.1. În cadrul gramaticilor de tip 1 un simbol neterminal x poate fi înlocuit printr-un cuvînt p dar numai în contextul $p_1 x p_2$. De aceea aceste gramatici se mai numesc *gramatici „context sensitive"* (*sensibile la context*), iar limbajele generate de ele se numesc și *limbaje „context sensitive"*.

În gramaticile de tip 2 un simbol neterminal x poate fi înlocuit cu un cuvînt p independent de contextul în care apare x. De aceea aceste gramatici se mai numesc și gramatici *„context free"* (*libere de context*), iar limbajele generate, *limbajele „context free"*.

Observația II.2.2. Are loc șirul de incluziuni stricte

$$\mathcal{L}_0 \supset \mathcal{L}_1 \supset \mathcal{L}_2 \supset \mathcal{L}_3 ,$$

așa cum vom arăta ulterior (v. teorema II.6.1 și exemplul II.6.1 și teoremele II.10.8 și II.15.3).

În continuare ne vom ocupa de unele proprietăți de închidere ale limbajelor față de operațiile de reuniune, produs și închidere prin produs a limbajelor (v. §1.3).

Teorema II.2.1. *Familiile $\mathcal{L}_j$ ($j = 0, 3$) conțin toate limbajele finite și sînt închise față de operația de reuniune.*

Demonstrație. Limbajul vid este generat de orice gramatică pentru care $I_T = \varnothing$. O astfel de gramatică este de orice tip. Limbajele finite sînt generate de gramatici similare acelcia descrise în exemplul II.1.2, care este de asemenea o gramatică de orice tip.

Fie gramaticile $G_j = (I_{Nj}, I_{Tj}, x_{0j} ; F_j)$, $j = 1, 2$ de tipul k și $L(G_j)$ limbajele generate. Putem presupune $I_{N1} \cap I_{N2} = \varnothing$, deoarece în caz contrar vom atribui cîte o etichetă distinctă fiecărui simbol din cele două alfabete.

Limbajul $L(G_1) \cup L(G_2)$ este generat de gramatica :

$$G_\cup = (I_N, I_{T1} \cup I_{T2}, y_0 : F_1 \cup F_2 \cup \{y_0 \to x_{01} , y_0 \to x_{02}\}),$$

unde

$$I_N = I_{N1} \cup I_{N2} \cup \{y_0\} \text{ și } y_0 \notin I_{N1} \cup I_{N2} .$$

Dacă gramaticile G_1 și G_2 sînt de tipul k, atunci și gramatica $G_\cup$ este de tipul k.

Avem $L(G_\cup) = L(G_1) \cup L(G_2)$. Într-adevăr dacă $p \in L(G_\cup)$ atunci $y_0 \underset{G_\cup}{\Rightarrow} p$. Or, aceasta presupune că în deducerea lui p a fost utilizată una din regulile de generare $y_0 \to x_{01}$ sau $y_0 \to x_{02}$. În primul caz $p \in L(G_1)$, iar în al doilea caz $p \in L(G_2)$. Deci $L(G_\cup) \subseteq L(G_1) \cup L(G_2)$. Apoi, dacă $p \in L(G_1) \cup L(G_2)$ avem sau $x_{01} \overset{*}{\underset{G_1}{\Rightarrow}} p$ sau $x_{02} \overset{*}{\underset{G_2}{\Rightarrow}} p$, de unde obținem $y_0 \overset{*}{\Rightarrow} p$, ținînd seama de prezența regulilor de generare $y_0 \to x_{01}$ sau $y_0 \to x_{02}$, deci $p \in L(G_\cup)$, adică $L(G_1) \cup L(G_2) \subseteq L(G_\cup)$. Astfel familia $\mathcal{L}_k$ este închisă la reuniune, $k = \overline{0, 3}$. ■

Teorema II.2.2. *Familia limbajelor libere de context, $\mathcal{L}_2$ este închisă la operațiile de produs și închidere prin produs.*

Demonstrație. Fie gramaticile de tip 2, $G_j = (I_{Nj}, I_{Tj}, x_{0j}; F_j)$, $j = 1, 2$ Limbajul $L(G_1)\, L(G_2)$ este generat de gramatica de tip 2:

$$G = (I_{N1} \cup I_{N2} \cup \{y_0\},\ I_{T1} \cup I_{T2},\ y_0;\ F_1 \cup F_2 \cup \{y_0 \to x_{01}\, x_{02}\}),$$

unde $y_0 \notin I_{N1} \cup I_{N2}$. Dacă $p \in L(G)$ atunci $y_0 \overset{*}{\underset{G}{\Rightarrow}} p$, ceea ce revine la: $y_0 \Rightarrow x_{01}\, x_{02} \overset{*}{\Rightarrow} p$. Atunci $p = p_1\, p_2$, unde $x_{01} \overset{*}{\underset{G_1}{\Rightarrow}} p_1$, $x_{02} \overset{*}{\underset{G_2}{\Rightarrow}} p_2$, adică $p \in L(G_1)\, L(G_2)$. Invers, dacă $p = p_1\, p_2$, unde $p_1 \in L(G_1)$, $p_2 \in L(G_2)$ avem $x_{01} \overset{*}{\underset{G_1}{\Rightarrow}} p_1$, $x_{02} \overset{*}{\underset{G_2}{\Rightarrow}} p_2$ și deci $y_0 \underset{G}{\Rightarrow} x_{01}\, x_{02} \overset{*}{\underset{G}{\Rightarrow}} p_1\, p_2$, adică $p_1\, p_2 \in L(G)$.

Limbajul $(L(G_1))^*$ este generat de gramatica de tip 2:

$$G_* = (I_{N1} \cup \{y_0\},\ I_{T1},\ y_0,\ F_1 \cup \{y_0 \to e,\ y_0 \to y_0\, x_{01}\}),$$

unde $y_0 \notin I_{N1}$.

Într-adevăr, dacă $p \in (L(G_1))^*$ atunci $p = q_1\, q_2 \ldots q_k$, unde $q_j \in L(G_1)$, deci $x_{01} \overset{*}{\Rightarrow} q_j$, $j = \overline{1, k}$.

Avem

$$y_0 \underset{G_*}{\Rightarrow} y_0\, x_{01} \underset{G_*}{\Rightarrow} y_0\, q_k \underset{G_*}{\Rightarrow} y_0\, x_{01}\, q_k \overset{*}{\underset{G_*}{\Rightarrow}} y_0\, q_{k-1}\, q_k \Rightarrow$$

$$\underset{G_*}{\Rightarrow} \ldots \underset{G_*}{\Rightarrow} y_0\, q_1\, q_2 \ldots q_k \underset{G_*}{\Rightarrow} q_1\, q_2 \ldots q_k = p,$$

adică $p \in L(G_*)$.

Invers, dacă $p \in L(G)^*$ trebuie să arătăm că $p = q_1\, q_2 \ldots q_t$, unde $x_{01} \overset{*}{\underset{G_1}{\Rightarrow}} q_j$, $j = \overline{1, t}$. Avem $y_0 \overset{*}{\underset{G_*}{\Rightarrow}} p$.

Plecînd de la simbolul inițial y_0 trebuie să aplicăm mai întîi, fie regula de generare $y_0 \to e$ fie regula de generare $y_0 \to y_0 x_{01}$.

În primul caz $p = e$ și $e \in (L(G_1))^*$. În al doilea caz:

$$y_0 \to y_0 x_{01} \overset{\bullet}{\Rightarrow} p.$$

Mai departe, în deducția $y_0 x_{01} \overset{\bullet}{\Rightarrow} p$ prima regulă de generare aplicată poate fi $y_0 \to e$, în care caz $p \in L(G_1)$, sau $y_0 \to y_0 x_{01}$. Apoi p poate aparține lui $(L(G_1))^2$, dacă în deducția $y_0 x_{01} x_{01} \overset{\bullet}{\Rightarrow} p$ prima regulă utilizată a fost $y_0 \to e$ sau lui $(L(G_1))^k$, $k > 2$, în caz contrar. ■

Teorema II.2.3. *Familia limbajelor de tipul* 3, $\mathcal{L}_3$ *este închisă față de operațiile de produs și închidere prin produs.*

Demonstrație. Fie $G_j = (I_{Nj}, I_{Tj}, x_{0j}; F_j)$, $j = 1, 2$ două gramatici de tip 3. Atunci gramatica de tip 3:

$$G = (I_{N1} \cup I_{N2},\ I_{T1} \cup I_{T2},\ x_{01},\ F' \cup F_2),$$

unde F' se obține din F_1 prin înlocuirea regulilor de generare de forma $x \to p$, cu $x \in I_{N1}$, $p \in I_{T1}$ prin regulile de generare $x \to px_{02}$, generează limbajul $L(G_1)\,L(G_2)$.

Dacă $p \in L(G_1)\,L(G_2)$ atunci $p = p_1 p_2$, unde $x_{01} \overset{\bullet}{\Rightarrow} p_1$, $x_{02} \overset{\bullet}{\Rightarrow} p_2$ Atunci în gramatica G, $x_{01} \overset{\bullet}{\Rightarrow} p_1 x_{02} \overset{\bullet}{\Rightarrow} p_1 p_2 = p$, deci $p \in L(G)$. Invers, dacă $p \in L(G)$ avem $x_{01} \overset{\bullet}{\Rightarrow} p$, adică $x_{01} \Rightarrow q_1 \Rightarrow q_2 \Rightarrow \ldots \Rightarrow q_n = p$. Fie q_k primul cuvînt de forma $p_2 x_{02}$. Atunci $p_1 \in L(G_1)$ și $p_1 x_{02} \overset{\bullet}{\Rightarrow} p_1 p_2 = p$, unde $p_2 \in L(G_2)$.

Apoi, gramatica de tip 3

$$G_* = (I_{N1} \cup \{y_0\},\ I_{T1}, y_0; F_1 \cup F'' \cup \{y_0 \to x_{01};\ x_{01} \to e\}),$$

unde $y_0 \notin I_{N1}$ și F'' se obține din F_2 înlocuind regulile de generare, de forma $x \to p$, $x \in I_{N1}$, $p \in I_{T1}^*$ prin regulile de generare de forma $x \to px_{01}$, generează limbajul $(L(G_1))^*$.

Dacă $p \in (L(G_1))^*$ atunci $p = q_1 q_2 \ldots q_k$, unde $q_j \in L(G_1)$, deci $x_{01} \overset{\bullet}{\underset{G_1}{\Rightarrow}} q_j$, $j = \overline{1, k}$.

Apoi $y_0 \underset{G_*}{\Rightarrow} x_{01} \overset{\bullet}{\underset{G_*}{\Rightarrow}} q_1 x_{01} \overset{\bullet}{\underset{G_*}{\Rightarrow}} q_1 q_2 x_{01} \overset{\bullet}{\underset{G_*}{\Rightarrow}} \ldots \overset{\bullet}{\underset{G_*}{\Rightarrow}} q_1 q_2 \ldots q_k x_{01} \overset{\bullet}{\underset{G_*}{\Rightarrow}} q_1 q_2 \ldots q_k$, aplicînd regula de generare $x_{01} \underset{G_*}{\to} e$, deci $(L(G_1))^* \subseteq L(G_*)$.

Să arătăm incluziunea inversă, $L(G_*) \subseteq (L(G_1))^*$.

Fie $p \in L(G_*)$. Trebuie să arătăm că $p = q_1 q_2 \ldots q_t$, unde $x_{01} \overset{\bullet}{\underset{G_*}{\Rightarrow}} q_j$, $j = \overline{1, t}$ sau $p = e$.

Relativ la mulțimea F_p a regulilor de generare aplicate pentru obținerea cuvîntului p distingem următoarele cazuri:

a) $F_p = \{y_0 \to x_{01};\ x_{01} \to e\}$; atunci $p = e$ și $p \in (L(G_1))^*$,

b) $F_p = F_{1p} \cup \{y_0 \to x_{01}\}$, unde $F_{1p} \subseteq F_1$; atunci $p \in L(G_1) \subseteq (L(G_1))^*$.

c) $F_p = F''_p \cup \{y_0 \to x_{01};\ x_{01} \to e\}$, unde $F''_p \subseteq F''$; atunci

$$y_0 \to x_{01} \underset{G_*}{\overset{\bullet}{\Rightarrow}} q_1 x_{01} \underset{G_*}{\overset{\bullet}{\Rightarrow}} q_1 q_2 x_{01} \underset{G_*}{\overset{\bullet}{\Rightarrow}} \dots \underset{G_*}{\overset{\bullet}{\Rightarrow}} q_1 q_2 \dots q_t x_{01} \underset{G_*}{\Rightarrow} q_1 q_2 \dots q_t \in$$
$$\in (L(G_1))^t \subseteq (L(G_1))^*.$$

d) $F_p = F_{1p} \cup F''_p \cup \{y_0 \to x_{01}, x_{01} \to e\}$, unde $F_{1p} \subseteq F_1$, $F''_p \subseteq F''$; atunci $y_0 \to x_{01} \underset{G_*}{\overset{\bullet}{\Rightarrow}} p$ și $p \in L(G) \subseteq (L(G_1))^*$ sau $y_0 \to x_{01} \underset{G_*}{\overset{\bullet}{\Rightarrow}} q_1 x_{01} \underset{G_1}{\overset{\bullet}{\to}} q_1 q_2 \in$ $\in L(G_1)^2$ sau $y_0 \to x_{01} \underset{G_*}{\overset{\bullet}{\Rightarrow}} q_1 x_{01} \underset{G_*}{\overset{\bullet}{\Rightarrow}} q_1 q_2 x_{01}$ ș.a.m.d., adică la fiecare pas putem aplica o regulă de deducție din F_{1p} sau din F''_p sau $x_{01} \to e$. În toate situațiile obținem un cuvînt $p \in (L(G_1))^t \subseteq (L(G_1))^*$.

Cazurile a) — d) epuizează toate posibilitățile, cum se constată imediat. ■

Teorema II.2.4. *Familia de limbaje $\mathcal{L}_0$ este închisă la operația de produs.*

Demonstrație. Se procedează la fel ca în cazul teoremei II.2.2 ■

Teorema II.2.5. *Familia de limbaje $\mathcal{L}_1$ este închisă la operația de produs.*

Demonstrație. Fie gramaticile $G_j = (I_{Nj}, I_{Tj}, x_{0j}; F_j)$, $j = 1, 2$ două gramatici de tip 1.

Să considerăm gramatica:

$$G = (I_{N1} \cup I_{N2} \cup \{y_0\},\ I_{T1} \cup I_{T2},\ y_0;\ F),$$

unde $y_0 \notin I_{N1} \cup I_{N2}$, $I_{N1} \cap I_{N2} = \varnothing$ (restricție care nu micșorează generalitatea), iar pentru F distingem următoarele cazuri:

a) $F = F_1 \cup F_2 \cup \{y_0 \to x_{01}\, x_{02}\}$ dacă $(x_{01} \to e) \notin F_1$ și $(x_{02} \to e) \notin F_2$

deoarece într-o gramatică sensibilă la context putem avea reguli de generare de forma $x \to e$ numai pentru simboli care nu apar în partea dreaptă a unei reguli de generare or, în gramatica G avem regula de generare $y_0 \to x_{01}\, x_{02}$.

b) $F = (F_2 \setminus \{x_{01} \to e\}) \cup F_2 \cup \{y_0 \to x_{01}\, x_{02};\ y_0 \to x_{02}\}$, dacă $(x_{02} \to e) \notin F_2$.

c) $F = F_1 \cup (F_2 \setminus \{x_{02} \to e\}) \cup \{y_0 \to x_{01}\, x_{02};\ y_0 \to x_{01}\}$, dacă $(x_{01} \to e) \notin F_1$.

d) $F = (F_1 \setminus \{x_{01} \to e\}) \cup (F_2 \setminus \{x_{02} \to e\}) \cup \{y_0 \to x_{01}\, x_{02};\ y_0 \to e\}$.

Gramatica G este de tipul 1, cum se constată imediat ținînd seama de definiția II.2.1.

Avem $L(G) = L(G_1)\, L(G_2)$, ceea ce se obține printr-un raționament identic cu cel folosit în teorema II.2.2. ■

Teorema II.2.6. *Familiile de limbaje $\mathcal{L}_0$ și $\mathcal{L}_1$ sînt închise la operația de închidere prin produs.*

Demonstrație. Fie gramatica $G = (I_N, I_T, x_0 ; F)$ de tipul j, $j = 0, 1$.

Limbajul $(L(G))^*$ este generat de gramatica de tipul $G_* = (I_N \cup \{y_0, y_1\}, I_T, y_0 ; F_*)$, unde $F_* = F_1 \cup \{y_0 \to e, y_0 \to x_0, y_0 \to x_0 y_1\} \cup \{iy \to ix_0, iy_1 \to ix_0 y_1, \forall i \in I_T\}$, unde :

$$F_1 = \begin{cases} F, \text{ dacă } j = 0 \\ F \setminus \{x_0 \to e\}, \text{ dacă } j = 1. \end{cases}$$

Avem $(L(G))^* \subseteq L(G_*)$. Într-adevăr, fie $p = p_1 p_2 \ldots p_k \in (L(G))^k \subseteq (L(G))^*$, adică $x_0 \overset{*}{\underset{G}{\Rightarrow}} p_j$, $j = \overline{1, k}$. În gramatica G_* avem deducția :

$$y_0 \to x_0 y_1 \overset{*}{\underset{G}{\Rightarrow}} p_1 y_1 \underset{G_*}{\Rightarrow} p_1 x_0 y_1 \overset{*}{\underset{G_*}{\Rightarrow}} p_1 p_2 y_1 \ldots \Rightarrow p_1 p_2 \ldots$$

$$\ldots p_{k-1} y_1 \overset{*}{\underset{G_*}{\Rightarrow}} \ldots \overset{*}{\underset{G_*}{\Rightarrow}} p_1 p_2 \ldots p_{k-1} x_0 \underset{G_*}{\Rightarrow} p_1 p_2 \ldots p_{k-1} p_k.$$

Invers, $L(G_*) \subseteq (L(G))^*$. Astfel, fie $p \in L(G_*)$. Prima regulă de generare aplicabilă în gramatica G_* poate fi :

1) $y_0 \to e$; în acest caz $p = e$ și $p \in (L(G))^*$

2) $y_0 \to x_0$, de aici rezultă $p \in L(G) \subseteq (L(G))^*$

3) $y_0 \to x_0 y_1$, dacă în gramatica G nu există deducția $x_0 \overset{*}{\underset{G}{\Rightarrow}} p$. În această situație $p = p_1 q$ unde $x_0 \overset{*}{\underset{G}{\Rightarrow}} p_1$. În continuare $y_0 \to x_0 y_1 \overset{*}{\underset{G_*}{\Rightarrow}} p_1 y_1 \overset{*}{\underset{G_*}{\Rightarrow}} p_1 p_2 y_1$, dacă $p = p_1 p_2 q'$ sau $y_0 \to x_0 y_1 \overset{*}{\underset{G_*}{\Rightarrow}} p_1 y_1 \overset{*}{\underset{G_*}{\Rightarrow}} p_1 x_0 \Rightarrow p_1 p_2$ dacă $p = p_1 p_2$ ș. a. m. d.

Procedeul se termină după un număr finit de pași, deoarece lungimea lui p este finită și obținem $p \in (L(G))^*$. ■

Teorema II.2.7. *Familiile de limbaje $\mathcal{L}_j$, $j = \overline{0,3}$ sînt închise la operațiile de reuniune, produs și închidere prin produs.*

Demonstrație. Rezultă din teoremele II.2.1 – II.2.6. ■

Observația II.2.3. Dacă, limbajul L este de tipul j atunci limbajele $L \cup \{e\}$ și $L \setminus \{e\}$ sînt de același tip j, ($j = \overline{0,3}$).

§ II. 3. AUTOMATE FINITE ȘI LIMBAJE REGULATE

În continuare ne vom ocupa de dispozitivele care „recunosc" limbajele generate de gramaticile de tipul 3, adică de automatele finite.

Definiţia II.3.1. Un *automat finit* este un 5-uplu $A = [I, S, O, f, g]$ unde I, S, O sînt mulţimi finite nevide numite respectiv *alfabet de intrare, mulţime de stări* şi *alfabet de ieşire*, $f: S \times I \to \mathcal{P}^*(S)$, $g: S \times I \to \mathcal{P}^*(O)$ sînt *funcţiile de trecere şi de ieşire* ale automatului.

Dacă $\forall s \in S, i \in I, |f(s, i)| = |g(s, i)| = 1$, atunci A se numeşte *automat determinist*. În caz contrar A este un *automat nedeterminist*.

Observaţia II.3.1. În toate cele ce urmează consideraţiile referitoare la noţiunea de automat sînt valabile atît pentru automatele deterministe cît şi pentru cele nedeterministe. Proprietăţile specifice unuia din tipurile de mai sus vor fi precizate ca atare. Pentru simplitatea expunerii un automat finit va fi numit pe scurt, automat.

Observaţia II.3.2. Noţiunea de automat finit intră în sfera mai largă a noţiunii de sistem cibernetic, care este un model al dispozitivelor de prelucrare a informaţiei provenind din cele mai diverse surse, funcţionînd la intervale de timp bine determinate numite *tacte*.

Ne putem imagina un automat ca o cutie neagră prevăzută cu o bornă de intrare şi o bornă de ieşire, capabilă să posede un număr finit de stări interne (elementele mulţimii S).

Aplicînd un semnal de intrare i (din alfabetul I) la borna de intrare sistemul respectiv elaborează un răspuns (un element din alfabetul O) şi trece într-o nouă stare internă. Acest răspuns precum şi starea în care trece automatul în tactul următor aparţin respectiv mulţimilor $g(s, i)$ şi $f(s, i)$, s fiind starea în tactul actual.

Există automate la care se impune ca starea în tactul iniţial al funcţionării lor să aparţină unei anumite mulţimi.

Definiţia II.3.2. Un *automat slab iniţial* este un 6-uplu $A = [I, S, O, f, g; S_0]$ unde I, S, O, f, g au semnificaţiile din definiţia II.3.1, $S_0 \subseteq S$ şi starea automatului în tactul iniţial aparţine lui S_0. Dacă $|S_0| = 1$, A este un *automat iniţial*; iar dacă $S_0 = S$, A este un *automat neiniţial*.

Definiţia II.3.3. Un *semiautomat* este un triplet $A = [I, S, f]$ unde I, S, f au semnificaţiile cunoscute. Noţiunea de *semiautomat slab iniţial* se introduce ca în definiţia precedentă.

Definiţia II.3.4. *Semiautomatul ataşat* automatului $A = [I, S, O, f, g]$ este semiautomatul $S(A) = [I, S, f]$.

Observaţia II.3.3. Prin considerarea semiautomatului $S(A)$ se face abstracţie de comportarea la ieşire a automatului A rămînînd esenţială funcţionarea internă a automatului.

Remarcăm că unui automat i se poate ataşa în mod unic un semiautomat, iar un semiautomat poate fi extins la un automat prin alegerea unui alfabet de ieşire O, şi a unei funcţii de ieşire g.

Automatul astfel ataşat nu este unic el depinzînd de această alegere.

Introducerea în n tacte succesive a cîte unui simbol $i_1, i_2, \ldots, i_n$ revine în fond la introducerea cuvîntului $p = i_1 \ldots i_n \in I^*$. Apare necesară astfel

extinderea funcţiilor de trecere şi de ieşire de la $S \times I$ la $S \times I^*$. Această extindere se va face în mod inductiv asupra lungimii cuvintelor din I şi anume :

$$f(s, e) = s, \tag{II.3.1}$$

$$f(s, pi) = f(f(s, p), i), \tag{II.3.2}$$

$$g(s, pi) = g(f(s, p), i). \tag{II.3.3}$$

Observaţia II.3.4. Se constată imediat că pentru $p = e$ din formulele (II.3. 1 – 3) obţinem :

$$\begin{aligned} f(s, ei) &= f(f(s, e), i) = f(s, i), \\ g(s, ei) &= g(f(s, e), i) = g(s, i), \end{aligned} \tag{II.3.4}$$

deci funcţiile extinse coincid pe $S \times I$ cu funcţiile automatului A.

Teorema II.3.1. *Fie* $A = [I, S, O, f, g]$. *Pentru* $\forall s \in S$, $i \in I$, $p, r \in I^*$ *au loc* :

$$f(s, ip) = f(f(s, i), p) \tag{II.3.5}$$

$$f(s, pr) = f(f(s, p), r) \tag{II.3.6}$$

$$g(s, ip) = g(f(s, i), p) \tag{II.3.7}$$

$$g(s, pr) = g(f(s, p), r). \tag{II.3.8}$$

Demonstraţie. Vom demonstra formula (II.3.6) prin inducţie asupra lungimii cuvîntului r, demonstrarea celorlalte relaţii făcîndu-se analog.

Pentru $l(r) = 0$, $r = e$, formula (II.3.6) este valabilă ţinînd seama de formula (II.3.1). Să presupunem formula adevărată pentru $l(r) = n$ şi să o demonstrăm pentru $q = ri$, $l(q) = n + 1$.

Conform formulei (II.3.2) şi ipotezei inductive se obţine :

$$f(s, pq) = f(s, pri) = f(f(s, pr), i) =$$

$$= f(f(f(s, p), r), i) = f(f(s, p), ri) = (f(f(s, p), q). \quad \blacksquare$$

Vom mai folosi funcţiile $f_s : I^* \to \mathcal{P}^*(S)$, $f_p : S \to \mathcal{P}^*(S)$, $g_s : I^* \to \mathcal{P}^*(O)$, $g_p : S \to \mathcal{P}^*(O)$ definite respectiv prin :

$$f_s(p) = f_p(s) = f(s, p),$$

$$g_s(p) = g_p(s) = g(s, p).$$

Pentru $S_1 \subseteq S$ şi $E \subseteq I^*$ vom mai folosi notaţia :

$$f(S_1, E) = \bigcup_{\substack{s \in S_1 \\ p \in E}} f(s, p), \quad g(S_1, E) = \bigcup_{\substack{s \in S_1 \\ p \in E}} g(s, p).$$

Fie $A = [I, S, O, f, g]$ un automat. Oricărei stări $s \in S$ îi corespunde o aplicaţie între semigrupurile libere I^* şi O^* aşa cum rezultă din :

Definiţia II.3.5. *Operatorul ataşat stării s* este aplicaţia $\gamma_s : I^* \to O^*$ definită prin

$$\gamma_s(i_1 i_2 \dots i_n) = g_s(i_1)\, g_{f(s,i_1)}(i_2) \dots g_{f(s,i_1,\dots,i_{n-1})}(i_n), \quad \forall i_1 i_2 \dots i_n \in I^*.$$

Exemplul II.3.1. Automatul $A = [\{i_1, i_2\}, \{s_1, s_2, s_3\}, \{o_1, o_2\}, f, g]$ unde f şi g sînt definite prin următoarele tabele :

f	s_1	s_2	s_3
i_1	s_2	s_1	s_3
i_2	s_3	s_2	s_2

g	s_1	s_2	s_3
i_1	o_1	o_2	o_1
i_2	o_2	o_1	o_2

este un automat determinist. Avem, de pildă $\gamma_{s_1}(i_1 i_2 i_1 i_2) = g_{s_1}(i_1)$.

$$g_{f(s_1,i_1)}(i_2)\; g_{f(s_1,i_1,i_2)}(i_1)\; g_{f(s_1,i_1,i_2,i_1)}(i_2) = o_1 o_1 o_1 o_2 .$$

Exemplul II.3.2. Automatul $A = [\{i_1, i_2\}, \{s_1, s_2\}, \{o_1, o_2\}, f, g]$ unde

f	s_1	s_2	s_3
i_1	$\{s_1, s_2\}$	$\{s_1\}$	$\{s_2\}$
i_2	$\{s_2\}$	$\{s_3\}$	$\{s_1\}$

g	s_1	s_2	s_3
i_1	$\{o_1\}$	$\{o_2\}$	$\{o_1, o_2\}$
i_2	$\{o_1, o_2\}$	$\{o_2\}$	$\{o_1\}$

este nedeterminist. Aici $\gamma_{s_1}(i_1 i_2 i_1 i_2) = \{o_1\}\{o_1, o_2\}\{o_1, o_2\}\{o_1, o_2\}$.

Ne ocupăm în continuare de limbajele care pot fi recunoscute de automatele finite *deterministe*.

Definiţia II.3.6. *Limbajul reprezentat în automatul iniţial* $A = [I, S, O, f_1 g; s_0]$ *prin semnalul de ieşire* $o \in O$ este mulţimea $E = g_{s_0}^{-1}(o) \subseteq I^*$, iar orice $p \in E$ va fi numit *cuvînt recunoscut de automatul A prin semnalul de ieşire o.*

Definiţia II.3.7. *Limbajul E este reprezentat în automatul iniţial* $A = [I, S, O, f, g; s_0]$ *prin mulţimea de stări* $S_1 \subseteq S$ *dacă* $E = f_{s_0}^{-1}(S_1)$.

Observaţia II.3.5. Definiţia II.3.7 se poate formula şi în cadrul mai restrîns al semiautomatelor iniţiale deoarece ea nu face apel la comportarea externă a automatului.

Legătura dintre definiţiile precedente este dată în :

Teorema II.3.2. *Fie* $E = g_{s_0}^{-1}(o)$. *Atunci pentru mulţimea de stări* $S_1 = f(s_0, E)$ *avem*

$$f_{s_0}^{-1}(S_1) = \begin{cases} E \cup \{e\}, & \text{dacă } s_0 \in S_1 \\ E, & \text{în caz contrar.} \end{cases}$$

Demonstraţie. Dacă $p \in E$, $f(s_0, p) \in S_1$ deci $p \in f_{s_0}^{-1}(S_1)$ adică $E \subseteq f_{s_0}^{-1}(S_1)$. Invers, fie $p \in f_{s_0}^{-1}(S_1)$. Atunci $f_{s_0}(p) \in S_1$ şi dacă $p \neq e$ urmează că $p \in E$. Dacă $s_0 \in S_1$ atunci $f_{s_0}(E) = s_0 \in S_1$ şi în acest caz $p \in E \cup \{e\}$. ■

Observația II.3.6. Teorema anterioară arată că reprezentarea limbajelor prin mulțimi de stări este mai generală decît reprezentarea lor prin semnale de ieșire deoarece în al doilea caz este exclus cuvîntul nul e.

În continuare, utilizînd numai reprezentarea limbajelor prin mulțimi de stări ne vom limita la semiautomate (v. observația II.3.5).

Limbajele reprezentabile în semiautomate finite vor fi numite *limbaje regulate*. Mulțimea limbajelor regulate din semigrupul liber I va fi notată cu $\mathcal{R}(I)$.

Pentru automatele nedeterministe definiția II.3.7 se extinde în mod natural.

Definiția II.3.8. Un *limbaj* $E \subseteq I^*$ *este reprezentat în semiautomatul nedeterminist slab inițial* $A = [I, S, f; S_0]$ prin mulțimea de stări S_1 dacă

$$E = \{p \mid f(S_0, p) \cap S_1 \neq \varnothing\}.$$

Teorema II.3.3. *Mulțimea limbajelor regulate coincide cu mulțimea limbajelor reprezentate în semiautomate nedeterministe slab inițiale.*

Demonstrație. Cum orice semiautomat inițial determinist poate fi interpretat ca un semiautomat nedeterminist slab inițial rezultă că orice limbaj regulat este reprezentabil în semiautomate nedeterministe.

Reciproc, fie E un limbaj reprezentat în semiautomatul nedeterminist slab inițial $A = [I, S, f; S_0]$ prin mulțimea de stări S_1. Să considerăm semiautomatul determinist inițial $A' = [I, \mathcal{P}^*(S), F; S_0]$ unde

$$F(S', i) = \bigcup_{s \in S'} f(s, i) \in \mathcal{P}^*(S)$$

Limbajul E coincide cu limbajul regulat E' reprezentat în semiautomatul A' prin mulțimea de stări $Z = \{S' \mid S' \cap S_1 \neq \varnothing\} \subseteq \mathcal{P}^*(S)$.

Într-adevăr, fie $p \in E$. Atunci $f(S_0, p) \cap S_1 \neq \varnothing$ și deci $F(S_0, p) = f(S_0, p) \in Z$, adică $p \in E'$. Dacă $p \in E'$, $F(S_0, p) \in Z$, adică $F(S_0, p) \cap S_1 \neq \varnothing$ și cum $f(S_0, p) = F(S_0, p)$ urmează $p \in E$. ■

Observația II.3.7. Utilitatea introducerii limbajelor reprezentate în semiautomate nedeterministe va reieși din paragraful următor unde le vom folosi în demonstrarea unor proprietăți de închidere pentru mulțimea $\mathcal{R}(I)$.

§ II. 4. LIMBAJE DE TIPUL 3

Teorema II.4.1. *Limbajele de tipul* 3 *coincid cu limbajele regulate.*

Demonstrație. Fie $G = (I_N, I_T, x_0; F)$ o gramatică de tip 3 unde $I_N = \{x_0, x_1, \ldots, x_n\}$. Regulile de generare în gramatica G pot fi fie de tipul $x_j \to p_k x_h$, fie de tipul $x_j \to p_k$, unde $x_j, x_h \in I_N$, $p_k \in I_T^*$. Să considerăm semiautomatul finit nedeterminist $A = [I_1, I_N \cup \{x, y\}, f; x_0]$, unde $I_1 = \{e\} \cup$

$\cup \{p_k \mid p_k \in I^*_T,\ \exists x_j, x_h \in I_N,\ (x_j \to p_k\, x_h) \in F$ sau $(x_j \to p_k) \in F\}$ cu alte cuvinte alfabetul de intrare I_1 constă din mulțimea finită a cuvintelor care apar în partea dreaptă a regulilor de generare din F la care se adaugă cuvîntul vid ; $x, y \notin I_N$.

Notînd $S_{jk} = \{x_h \mid (x_j \to p_k\, x_h) \in F\}$ definim f prin

$$f(x_j, p_k) = \begin{cases} \{x\} \cup S_{jk}, \text{ dacă } (x_j \to p_k) \in F \\ y, \text{ dacă } (x_j \to p_k) \notin F \text{ și } S_{jk} = \varnothing \end{cases}$$

$$f(x, p_k) = f(y, p_k) = y, \quad \forall p_k \in I_1.$$

$$f(x_0, e) = x_0.\ f(x_j, e) = y,\ f(y, e) = y.$$

Vom arăta că limbajul $L(G)$ coincide cu limbajul E, reprezentat în semiautomatul A prin mulțimea de stări S_1, unde :

$$S_1 = \begin{cases} \{x_0, x\}, \text{ dacă } (x_0 \to e) \in F \\ \{x\}, \text{ în caz contrar.} \end{cases}$$

Putem presupune, fără a restrînge generalitatea că dacă $(x_0 \to e) \in F$, atunci x_0 nu apare în partea dreaptă a vreunei reguli de generare din F, deoarece, dacă avem simultan $(x_0 \to e)$, $(x_j \to px_0) \in F$, putem înlocui regula de generare $(x_j \to px_0)$ prin $(x_j \to p)$, fără a schimba tipul gramaticii și limbajul generat de ea.

Fie $p = p_1 p_2 \dots p_n \in L(G)$, $n \geqslant 1$. Atunci avem deducția $x_0 \Rightarrow p_1 x_{j_1} \Rightarrow \dots \Rightarrow p_1 p_2 \dots p_{n-1} x_{j_{n-1}} \Rightarrow p_1 p_2 \dots p_n$ și în semiautomatul A obținem șirul de stări : $x_{j_1} \in f(x_0, p_1)$, $x_{j_2} \in f(x_{j_1}, p_2), \dots, x \in f(x_{j_{n-1}}, p_n)$, adică $x \in f(x_0, p)$, sau $p \in E$. Dacă $e \in L(G)$, atunci $(x_0 \to e) \in F$, $x_0 \in S_1$ și $e \in E$, de unde $L(G) \subseteq E$. Să arătăm acum că $E \subseteq L(G)$. Fie $p \in E$ cu $p \neq e$. Există atunci un șir de stări $x_0, x_{j_1}, x_{j_2}, \dots, x_{j_{n-1}}, x$, astfel ca $x_{j_1} \in f(x_0, p_1)$, $x_{j_2} \in f(x_{j_1}, p_2), \dots$ $\dots x_{j_{n-1}} \in f(x_{j_{n-2}}, p_{n-1})$, $x \in f(x_{j_{n-1}}, p_n)$, deci există în F regulile de generare $x_0 \to p_1 x_{j_1}$, $x_{j_1} \to p_2 x_{j_2}, \dots, x_{j_{n-1}} \to p_n$, adică există deducția $x_0 \overset{*}{\underset{G}{\Rightarrow}} p_1 p_2 \dots p_n$, deci $p \in L(G)$. Dacă $p = e \in E$, atunci $f(x_0, e) = x_0 \in S_1$, de unde $(x_0 \to e) \in F$ și $e \in L(G)$.

Reciproc, fie E limbajul reprezentat în semiautomatul determinist $A = [I, S, f ; s_0]$ prin mulțimea de stări S_1. Definim gramatica de tip 3, $G = (S, I, s_0 ; F)$, unde

$$F = \{(s_j \to p_k s_h) \mid s_h = f(s_j, p_k)\} \cup \{(s_j \to p_k) \mid f(s_j, p_k) \in S_1\}.$$

Să arătăm că $E = L(G)$. Fie $p = i_1 i_2 \dots i_n \in E$, $p \neq e$. Există șirul de stări $s_0, s_1, \dots, s_n$, unde $s_1 = f(s_0, i_1)$, $s_2 = f(s_1, i_2), \dots, s_n = f(s_{n-1}, i_n)$, $s_n \in S_1$. Corespunzător, în gramatica G apar regulile de generare $s_0 \to i_1 s_1$, $s_1 \to i_2 s_2, \dots$ $\dots, s_{n-1} \to i_n$ și astfel există deducția $s_0 \overset{*}{\underset{G}{\Rightarrow}} i_1 i_2 \dots i_n = p$. Prin urmare, $p \in L(G)$. Dacă $p = e$, atunci $s_0 \in S_1$ și avem regula de generare $s_0 \to e$, adică

$e \in L(G)$. Astfel $E \subseteq L(G)$. Invers, fie acum $p \in L(G)$, $p \neq e$. Cuvîntul p se obține atunci aplicînd succesiv regulile de generare de forma :

$s_0 \to i_1 s_1 \quad s_1 \to i_2 s_2 \ldots, s_{n-1} \to i_n$, ceea ce antrenează pentru automatul A existența șirului de stări $s_0 \; s_1, \ldots, s_{n-1}, s_n$ cu $s_n = f(s_{n-1}, i_{n-1}) \in S_1$ și $s_h = f(s_{h-1}, i_h)$, $h = \overline{1, n-1}$.

Atunci $s_n = f(s_0, i_1 i_2 \ldots i_{n-1}) \in S_1$ și $p \in E$. Dacă $p = e$, atunci p a fost obținut aplicînd regula de generare $s_0 \to e$, de unde $f(s_0, e) = s_0 \in S_1$ și deci $e \in E$. ■

Exemplul II.4.1. Să considerăm gramatica de tip 3, $G = (\{x_0, x_1\}, \{i_1, i_2\}, x_0; F)$, unde $F = \{x_0 \to i_1 x_1, x_1 \to i_1 x_1, x_1 \to i_2 x_0, x_1 \to i_1\}$. În semiautomatul finit determinist $A = [I_1, \{x_0, x_1, x, y\}, f; x_0]$, unde $I_1 = \{e, i_1, i_2\}$, iar f este dat de tabloul :

f	x_0	x_1	x	y
e	$\{x_0\}$	$\{y\}$	$\{y\}$	$\{y\}$
i_1	$\{x_1\}$	$\{x, x_1\}$	$\{y\}$	$\{y\}$
i_2	$\{y\}$	$\{x_0\}$	$\{y\}$	$\{y\}$

limbajul $L(G)$ este reprezentat de mulțimea de stări $\{x\}$. Avem, de exemplu, $f(x_0, i_1 i_1) \cap \{x\} \neq \varnothing$, deci $i_1 i_1 \in L(G)$. Apoi $i_1 i_2 i_1 i_1 \in L(G)$, etc.

Teorema II.4.2. *Mulțimea limbajelor regulate este închisă la operațiile de reuniune, produs și închidere prin produs.*

Demonstrație. Se ține seama de teoremele II.4.1, II.2.1, II.2.3. ■

Teorema II.4.3. *Fie $E \subseteq I^*$. Următoarele afirmații sînt echivalente :*

1) *E este un limbaj regulat, $E \in \mathcal{R}(I)$*

2) *E este o reuniune de clase de echivalență ale unei congruențe de rang finit pe I^*.*

3) *congruența μ^E este de rang finit, unde*

$$\mu^E = \{(p, q) \mid \chi_E(r_1 p r_2) = \chi_E(r_1 q r_2), \; \forall r_1, r_2 \in I^*\}.$$

Demonstrație. Vom arăta că au loc următoarele implicații : 1) $\Rightarrow$ 2), 2) $\Rightarrow$ 3), 3) $\Rightarrow$ 1).

1) $\Rightarrow$ 2). Fie $E \in \mathcal{R}(I)$ și $A = [I, S, f; s_0]$ un semiautomat inițial, finit, în care E este reprezentat prin mulțimea de stări S_1, adică $p \in E \leftrightarrow f(s_0, p) \in S_1$.

Definim relația $\zeta = \bigcap_{s \in S} \ker f_s$. ζ este o congruență pe I^*. ζ este o echivalență, fiind intersecția unei mulțimi de relații de echivalență. Fie acum $(p, q) \in \zeta$. Să verificăm că $(pr, qr) \in \zeta$, $(rp, rq) \in \zeta$, $\forall r \in I^*$. Avem $f_s(rp) = (f(f(s, r), p) = f(f(s, r), q) = f_s(rq)$, $\forall s$; la fel $f_s(pr) = f(f(s, p), r) = f(f(s, q), r) = f_s(qr)$, $\forall s$, adică ζ este o congruență.

Relația ζ este de rang finit. Fie $S = \{s_1, \ldots, s_k\}$ și o aplicație $\tau : S \to S$. Notăm $I^*(\tau) = \{p \mid p \in I^*, f(s_h, p) = \tau(s_h) ; h = \overline{1, k}\}$.

Să arătăm că $(p, q) \in \zeta$ dacă și numai dacă există o aplicație τ astfel ca $\{p, q\} \subseteq I^*(\tau)$.

Fie $(p, q) \in \zeta$. Definim aplicația τ prin $\tau(s_h) = f(s_h, p) = f(s_h, q)$. Avem $p \in I^*(\tau)$ și $q \in I^*(\tau)$. Invers, dacă $\{p, q\} \subseteq I^*(\tau)$ atunci $f(s_h, p) = \tau(s_h) = f(s_h, q)$, deci $(p, q) \in \zeta$. Astfel, fiecărei clase din mulțimea I^*/ζ i se atașează în mod biunivoc o aplicație $\tau : S \to S$. Deoarece numărul acestor aplicații nu poate depăși k^k, urmează $|I^*/\zeta| \leqslant k^k$, adică ζ este o congruență de rang finit.

Fie acum $p \in E$ și $(p, q) \in \zeta$. Atunci $f(s_0, q) = f(s_0, p) \in S_1$, deci $q \in E$. Astfel E este reuniunea unor clase de echivalență ale lui ζ.

2) $\Rightarrow$ 3). Fie ζ o congruență de rang finit pe I^* și E o reuniune de clase de echivalență după relația ζ. Fie $(p, q) \in \zeta$. Dacă $\chi_E(r_1 pr_2) = 1$, atunci, deoarece $(r_1 pr_2, r_1 qr_2) \in \zeta$, obținem $\chi_E(r_1 qr_2) = 1$, deci $(p, q) \in \mu^E$. Prin urmare, $\zeta \subseteq \mu^E$, de unde $|I^*/\mu^E| \leqslant |I^*/\zeta|$.

3) $\Rightarrow$ 1). Fie E conținut în I^* și să presupunem că $|I^*/\mu^E| < \infty$. Limbajul E este reprezentat în semiautomatul inițial $A = [I, I^*/\mu^E, f; [e]]$, unde $f([p], i) = [pi]$, prin mulțimea de stări $S_1 = \{[p] \mid p \in E\}$. Se observă că $f([p], q) = [pq]$.

Avem $p \in E \leftrightarrow [p] \in S_1 \leftrightarrow f([e], p) \in S_1$. ■

Teorema II.4.4. Fie $E \subseteq I^*$. *Următoarele afirmații sînt echivalente*:

1) *E este un limbaj regulat, $E \in \mathcal{R}(I)$*

2) *E este o reuniune de clase după o congruență la dreapta pe I^*, de rang finit.*

3) *Congruența la dreapta ν^E este de rang finit, unde*

$$\nu^E = \{(p, q), \chi_E(pr) = \chi_E(qr), \forall r \in I^*\}.$$

Demonstrație. Se procedează analog, ca la teorema II.4.3, definind încă, $\zeta = \ker f_{s_0}$; se constată imediat că ζ este o congruență la dreapta. ■

Teorema II.4.5. *Dacă $E \in \mathcal{R}(I)$, atunci numărul minim de stări ale unui semiautomat în care E este reprezentabil este egal cu $|I^*/\nu^E|$.*

Demonstrație. Procedăm prin reducere la absurd. Să presupunem că limbajul E poate fi reprezentat în semiautomatul $A' = [I, S', f'; s'_0]$ și $|S'| < |I^*/\nu^E|$.

Să considerăm relația $\zeta' = \ker f'_{s'_0}$. Ca în demonstrarea implicației 2) $\Rightarrow$ 3) din teorema II.4.3 se arată că $|I^*/\nu^E| \leqslant |I^*/\zeta'|$.

Aplicația $\tau : I^*/\zeta' \to S'$ definită prin $\tau([p]) = f'_{s'_0}(p)$ este injectivă, deci $|I^*/\zeta'| \leqslant |S'|$, ceea ce contrazice presupunerea inițială. ■

Teorema II.4.6. *$E \in \mathcal{R}(I)$ dacă și numai dacă $E^c \in \mathcal{R}(I)$, unde $E^c = I^* \setminus E$, adică $\mathcal{R}(I)$ este închis la complementariere.*

Demonstrație. Dacă $E = f_{s_0}^{-1}(S_1)$, atunci $E^c = f_{s_0}^{-1}(S \setminus S_1)$, unde $A = [I, S, f; s_0]$ este semiautomatul în care E este reprezentat prin mulțimea de stări S_1. ■

Teorema II.4.7. *$\mathcal{R}(I)$ este închis la intersecție.*

Demonstrație. Fie $E_1, E_2 \in \mathcal{R}(I)$ și $A_j = [I, S^j, f^j; s_0^j]$, $j = 1, 2$ semiautomatele în care limbajele E_1 și E_2 sînt reprezentate respectiv prin mulțimile de stări $S_1 \subseteq S^1$, $S_2 \subseteq S^2$. Să considerăm semiautomatul $A = [I, S^1 \times S^2, f; (s_0^1, s_0^2)]$, unde $f: S^1 \times S^2 \times I \to S^1 \times S^2$, astfel ca $\mathrm{pr}_j f = f_j$, $j = 1, 2$.

Limbajul $E_1 \cap E_2$ este reprezentat în semiautomatul A prin mulțimea de stări $S_1 \times S_2 \subseteq S^1 \times S^2$. Într-adevăr, dacă $p \in E_1 \cap E_2$, atunci $f^1(s_0^1, p) \in S_1$, $f^2(s_0^2, p) \in S_2$, deci $f((s_0^1, s_0^2), p) \in S_1 \times S_2$ și reciproc, prin urmare, $E_1 \cap E_2 \in \mathcal{R}(I)$. ■

Observația II.4.1. Ținînd seama de teoremele II.2.1, II.4.1, II.4.6, se obține imediat rezultatul precedent, deoarece $E_1 \cap E_2 = (E_1^c \cup E_2^c)^c$.

Teorema II.4.8. *Mulțimea $\mathcal{R}(I)$ este algebră Boole față de operațiile de reuniune, intersecție și complementariere de mulțimi.*

Demonstrație. Rezultă imediat din teoremele precedente. ■

Putem regăsi un rezultat cuprins în teorema II.2.1 pentru cazul $j = 3$, utilizînd exclusiv semiautomatele finite:

Teorema II.4.9. *Dacă $|E| < +\infty$, atunci $E \in \mathcal{R}(I)$.*

Demonstrație. Este suficient să arătăm că dacă $|E| = 1$, adică $E = \{p\}$, atunci $E \in \mathcal{R}(I)$ și să ținem seama de teorema II.4.8.

Dacă $p = e$, atunci E poate fi reprezentat într-un semiautomat inițial oarecare $A = [I, S, f; s_0]$ prin starea s_0, f fiind ales astfel, încît $f(s, i) \neq s_0$, $\forall i \in I$, A $\forall s \in S$.

Fie $p = i_1 i_2 \ldots i_n$. Atunci limbajul E va fi reprezentat în semiautomatul $A = [I, \{e, i_1, i_1 i_2, \ldots, i_1 i_2 \ldots i_n, s\}, f; e]$, unde

$$f(i_1 \ldots i_j, i) = \begin{cases} i_1 \ldots i_j i_{j+1}, \text{ dacă } i = i_{j+1} \\ s, \text{ în caz contrar;} \end{cases}$$

$$f(e, i) = \begin{cases} i_1, & \text{dacă } i = i_1 \\ s, & \text{dacă } i \neq i_1, \end{cases}$$

$f(s, i) = f(i_1 \ldots i_n, i) = s$, prin starea $i_1 i_2 \ldots i_n$. ■

Aceluiași scop, și anume regăsirea proprietăților limbajelor regulate cu ajutorul semiautomatelor finite, îi sînt consacrate următoarele două teoreme:

Teorema II.4.10. *Mulțimea limbajelor regulate este închisă la produs.*

Demonstrație. Fie $E_1, E_2 \in \mathcal{R}(I)$ și semiautomatele $A_j = [I, S^j, f^j, s_0^j]$, $j = 1, 2$, în care limbajele E_1 și E_2 sînt reprezentate respectiv prin mulțimile de stări $S_1 \subseteq S^1$, $S_2 \subseteq S^2$.

Vom arăta că limbajul $E_1 E_2$ este reprezentat în semiautomatul nedeterminist $A = [I, S^1 \cup' S^2, f; s_0^1]$ [1] prin mulțimea de stări S_2.

Aici:

$$f(s, i) = \begin{cases} \{f^1(s, i)\}, \text{ dacă } s \in S^1 \setminus S_1 \\ \{f^1(s, i), f^2(s_0^2, i)\}, \text{ dacă } s \in S_1 \\ \{f^2(s, i)\}, \text{ dacă } s \in S^2. \end{cases}$$

[1] S-a notat prin $\cup'$ reuniunea disjunctă.

Să notăm cu E evenimentul reprezentat în semiautomatul A prin mulțimea de stări S_2 și fie $p_1 \in E_1$, $p_2 \in E_2$. Avem $f^1(s_0^1, p_1) \in S_1$ și, dacă $p_2 = i_1 \dots i_n$, atunci $f^2(s_0^2, i_1) \in f(f^1(s_0^1, p_1), i_1)$. De aici rezultă imediat că $f(s_0^1, p_1 p_2) \cap S_2 \neq \emptyset$, adică $p_1 p_2 \in E$, deci $E_1 E_2 \subseteq E$.

Invers, fie $p = i_1 \dots i_n \in E$. Atunci $f(s_0^1, p) \cap S_2 \neq \emptyset$, și fie $k = \min\{j \mid f(s_0^1, i_1 \dots i_{j+1}) \neq f^1(s_0^1, i_1 \dots i_{j+1})\}$. Din modul de definire a semiautomatului A, cuvîntul $p_1 = i_1 \dots i_k \in E_1$, iar cuvîntul $p_2 = i_{k+1} \dots i_n \in E_2$, adică $E \subseteq E_1 E_2$. ■

Teorema II.4.11. *Mulțimea limbajelor regulate este închisă la operația de închidere prin produs.*

Demonstrație. Să presupunem că limbajul E este reprezentat în semiautomatul determinist $A = [I, S, f; s_0]$ prin mulțimea de stări S_1 și fie semiautomatul nedeterminist $A' = [I, S, F; s_0]$, unde:

$$F(s, i) = \begin{cases} \{f(s, i)\}, \text{ dacă } s \in S \setminus S_1 \\ \{f(s, i), f(s_0, i)\}, \text{ dacă } s \in S_1. \end{cases}$$

Fie E' limbajul reprezentat în A' prin mulțimea de stări S_1. Să arătăm că $E^* = E'$.

Fie $p \in E^*$. Atunci există $k \in N$, $\{q_1, q_2, \dots, q_k\} \subseteq E$ și $p = q_1 q_2 \dots q_k$. Dar, $f(s_0, q_1) \in S_1$ și ca atare, dacă $q_2 = i_1 \dots i_m$, $F(s_0, q_1 i_1) = \{f(s_0, q_1 i_1), f(s_0, i_1)\}$. Se vede imediat că: $F(s_0, q_1 q_2) \cap S_1 \neq \emptyset$ și, mai general, $F(s_0, q_1 q_2 \dots q_h) \cap S_1 \neq \emptyset$, pentru $h = \overline{1, k}$. În particular, pentru $h = k$ obținem $F(s_0, p) \cap S_1 \neq \emptyset$, deci $p \in E'$.

Invers, fie $p = i_1 i_2 \dots i_n \in E'$. Avem $F(s_0, p) \cap S_1 \neq \emptyset$. Se vede ușor că p se poate scrie $p = q_1 q_2 \dots q_k$, $k \geq 1$, unde $f(s_0; q_h) \in S_1$, $\forall h = \overline{1, k}$ și deci $p \in E^k \subseteq E^*$. ■

Semnalăm, în continuare, alte proprietăți de închidere ale mulțimii $\mathcal{R}(I)$.

Definiția II.4.1. *Operația de oglindire* este operația unară în I^*, $\sim : I^* \to I^*$ dată de: $\tilde{p} = i_n i_{n-1} \dots i_1$, unde $p = i_1 i_2 \dots i_n$.

Observația II.4.2. Operația de oglindire se extinde la $\mathcal{P}(I^*)$ punînd $\tilde{E} = \{\tilde{p} \mid p \in E\}$.

Avem $\tilde{\tilde{E}} = E$ și $\widetilde{(pq)} = \tilde{q}\,\tilde{p}$.

Teorema II.4.12. *$\mathcal{R}(I)$ este închisă la operația de oglindire.*

Demonstrație. Fie $E \in \mathcal{R}(I)$, adică $|I^*/\mu^E| < \infty$, conform teoremei II.4.3. Vom arăta că $|I^*/\mu^{\tilde{E}}| < +\infty$, ceea ce va antrena $\tilde{E} \in \mathcal{R}(I)$.

Fie $(p, q) \in \mu^E$. Atunci $\forall r_1, r_2 \in I^*$, $\chi_E(r_1 p r_2) = \chi_E(r_1 p r_2)$ sau $\chi_{\tilde{E}}(r_3 \tilde{p} r_4) = \chi_{\tilde{E}}(r_3 \tilde{q} r_4)$, $\forall r_3, r_4 \in I^*$. Deci $(\tilde{p}, \tilde{q}) \in \mu^{\tilde{E}}$. ■

Definiția II.4.2. *Operația de închidere prin prefixe* este aplicația $\mathbf{pr} : \mathcal{P}(I^*) \to \mathcal{P}(I^*)$ dată de

$$\mathbf{pr}(E) = \{q \mid \exists r \in I^*, qr \in E\} \cup \{e\}.$$

Lăsăm în seama cititorului verificarea proprietăților de operator de închidere (v. definiția I.1.13) ale aplicației **pr**.

Teorema II.4.13. *$\mathcal{R}(I)$ este închisă la operația* **pr**.

Demonstrație. Fie $E \in \mathcal{R}(I)$ și $A = [I, S, f; s_0]$ semiautomatul în care limbajul E este reprezentat prin mulțimea de stări S_1. Se constată imediat că evenimentul **pr** E este reprezentat în automatul A prin mulțimea de stări $S' = \{s \mid \exists p \in E,\ p = qr,\ f(s_0, q) = s\}$. ■

Teorema II.4.14. *Fie $E \in \mathcal{R}(I)$ și $A = [I, S, f; s_0]$ semiautomatul în care E este reprezentat. Este decidabil faptul că $E \neq \varnothing$.*

Demonstrație. Să presupunem că $E \neq \varnothing$. Vom arăta că există un cuvînt $p \in E$, $l(p) < |S|$, ceea ce ne va permite ca într-un număr finit de pași (cel mult $|I^*|^n$, $n = |S|$). Să verificăm dacă $E \neq \varnothing$.

Fie $p \in E$ cu $l(p) = \min\{l(q) \mid q \in E\}$. Dacă $l(p) \geqslant |S|$, atunci fie $s_{k+1} = f(s_k, i_k)$, $k = \overline{0, l(p)}$, unde $p = i_1 i_2 \ldots i_n$. Mulțimea $\{s_{k+1} \mid k = \overline{0, l(p)}\}$, avînd cel puțin $|S| + 1$ elemente nu poate avea toate elementele distincte. Există deci j_1, j_2 astfel ca $s_{j_1+1} = s_{j_2+1}$, $0 \leqslant j_1 < j_2 \leqslant l(p)$. Atunci cuvîntul $q = i_1 \ldots i_{j_1} i_{j_2} \ldots i_{l(p)} \in E$ și $l(q) < l(p)$, ceea ce contrazice definiția lui p. ■

Fie $E_1, \ldots, E_r$ limbaje peste alfabetul $I = \{i_1, \ldots, i_r\}$. Fie aplicația $\varphi : I^* \to \mathcal{P}(I^*)$ dată de

$$\varphi(e) = \{e\},$$
$$\varphi(i_j) = E_j,\ j = \overline{1, r},$$
$$\varphi(pq) = \varphi(p)\varphi(q),\ \forall p, q \in I^*.$$

Definiția II.4.3. *Operația de substituție generată de mulțimea de limbaje* $M = \{E_1, \ldots, E_r\}$ este operația unară $\Phi_M : \mathcal{P}(I^*) \to \mathcal{P}(I^*)$ dată de:

$$\Phi_M(E) = \{q \mid q \in \varphi(p),\ p \in E\}.$$

Operația de substituție ψ este o operație $(r+1)$-ară pe $\mathcal{P}(I^*)$ dată de:

$$\psi(E_1, \ldots, E_r, E) = \Phi_M(E), \text{ unde } M = \{E_1, \ldots, E_r\}.$$

Observația II.4.3. Dacă $E_j = \{i_{\alpha(j)}\}$, unde $\alpha : \overline{1, r} \to \overline{1, r}$, atunci aplicația φ este un endomorfism pe I^* care păstrează lungimea cuvintelor și $\Phi_M(E) = \varphi(E)$.

Teorema II.4.15. *$\mathcal{R}(I)$ este închisă la operația de substituție.*

Demonstrație. Fie $E_1, \ldots, E_r$, $E \in \mathcal{R}(I)$ și $E_j = L(G_j)$, unde $G_j = (I_N^j, \{i_1, \ldots, i_r\}, x_0^j, F^j)$, $j = \overline{1, r}$, și $E = L(G)$, cu $G = (I_N, \{i_1, \ldots, i_r\}, x_0; F)$, toate gramaticile care intervin aici fiind de tip 3.

Să considerăm gramatica de tipul 3, $G' = (I_N \cup \bigcup_{j=1}^{r} I_N^j, \{i_1, \ldots, i_r\}, x_0; F')$, unde $F' = \bigcup_{j=1}^{r} \bar{F}_j \cup \bar{F}$.

Aici $\bar{F}_j$ se obține din F_j, înlocuind regulile de generare de forma $x_k^j \to i_1, i_2 ... i_h$ prin reguli de forma $x_k^j \to i_1 i_2 ... i_h x_0^l$, dacă în F apare o regulă de forma $x \to ... i_h i_l ... x..$ și $x_k^j \to i_1 i_2 ... i_h x_l$, dacă în G avem regulile de generare $x \to ... i_h x_l$, $x_l \to ...$

$\bar{F}$ se obține din F înlocuind regulile de forma $x_l \to i_{j_1} ... i_{j_n} x_j$ sau $x_l \to i_{j_1} i_{j_2} ... i_{j_n}$ cu reguli de forma $x_l \to x_0^j l$.

Să arătăm că $L(G') = \psi(E_1, ..., E_r, E)$. Fie $p \in \psi(E_1, ..., E_r, E)$. Atunci $p = p_1 ... p_h$, unde $p_k \in E_{j_k}$, $\forall k = \overline{1, h}$ și $i_{j_1} i_{j_2} ... i_{j_h} \in E$.
Există atunci derivarea:
$x_0 \Rightarrow i_{j_1} ... i_{j_{k_1}} x_1 \Rightarrow i_{j_1} ... i_{j_{k_1}} ... i_{j_{k_2}} x_2 \Rightarrow ... \Rightarrow i_{j_1} i_{j_2} ... i_{j_h}$ prevenită prin aplicarea regulilor de generare din gramatica G:
$x_0 \to i_{j_1} i_{j_2} ... i_{j_{k_1}} x_1$, $x_1 \to i_{j_{k_1+1}} ... i_{j_{k_2}} x_2, ..., x_s \to i_{j_{k_s+1}} ... i_{j_h}$. Corespunzător, în gramatica G' avem regulile de generare:
$x_0 \to x_0^{j_1}$ $x_1 \to x_0^{j_{k_1+1}} ... x_s \to x_0^{j_{k_s+1}}$ din $\bar{F}$ și, cum $p_1 \in E_{j_1}, ..., p_h \in E_{j_h}$ apare derivarea:

$$x_0^{j_1} \underset{G'}{\overset{*}{\Rightarrow}} p_1 x_0^{j_2} \underset{G'}{\overset{*}{\Rightarrow}} ... \underset{G'}{\overset{*}{\Rightarrow}} p_1 ... p_{j_{k_1}} x_1 \underset{G'}{\overset{*}{\Rightarrow}} p_1 ... p_{j_{k_1}} p_{j_{k_1+1}} x_0^{j_{k_1+2}} \underset{G'}{\overset{*}{\Rightarrow}} ... \underset{G'}{\overset{*}{\Rightarrow}} p_1 ... p_{j_{k_1}} ...$$

$$... p_{j_{k_2}} x_2 \underset{G'}{\overset{*}{\Rightarrow}} ... \underset{G'}{\overset{*}{\Rightarrow}} p_1 ... p_{j_{k_1}} ... p_h, \text{ deci } p \in L(G').$$

Invers, fie $p \in L(G')$. În gramatica G' există în definitiv, cinci tipuri de reguli de generare: $x_k \to x_0^j$, $x_k^j \to q x_0^l$, $x_k^j \to r x_l$, $x_k^j \to p$, $x_k^j \to p x_1^j$. Să presupunem că în generarea cuvîntului p apar toate aceste tipuri de reguli de generare. Fie, de exemplu,:
$x_0 \Rightarrow x_0^1 \to q' x_0^2 \Rightarrow q' q'' x_0^3 \Rightarrow q' q'' r x_4 \Rightarrow q' q'' r x_0^5 \Rightarrow q' q'' r r' x_2^5 \Rightarrow q' q'' r r' p x_3^5 \Rightarrow$
$\Rightarrow q' q'' r r' p p'$.
Cum în acest caz $pp' \in E_{j_s}$, rezultă că p este obținut prin substituția ψ efectuată asupra cuvîntului $i_1 i_2 i_3 i_4 i_5 \in E$. Astfel $p \in \psi(E_1, ..., E_r, E)$. ■

Teorema II.4.16. *$\mathcal{R}(I)$ este închis la endomorfismele semigrupului liber I^*.*

Demonstrație. Rezultă din observația II.4.3 și din teorema II.4.15. ■

Teorema II.4.17. *Dacă $E \in \mathcal{R}(I)$ și h este un endomorfism al semigrupului liber I^*, care păstrează lungimea cuvintelor, atunci $h^{-1}(E) \in \mathcal{R}(I)$.*

Demonstrație. Aplicația $h^{-1}: I^* \to \mathcal{P}(I^*)$ face să corespundă unui simbol $i \in I$, mulțimea finită $h^{-1}(i) \subseteq I$. Atunci $h^{-1}(i)$ este un limbaj regulat (v. teorema II.2.1) și $h^{-1}(E) = \psi(h^{-1}(i_1), ..., h^{-1}(i_r), E)$. ■

În sfîrșit, limbajele regulate apar ca soluții ale acelor sisteme de ecuații liniare cu coeficienți limbaje regulate, care admit soluție unică. Mai precis, are loc:

Teorema II.4.18. *Sistemul $Z = ZE \cup B$, în care E este o matrice nesingulară ale cărei elemente sînt limbaje regulate și B este un vector-linie cu compo-*

nente limbaje regulate, are ca soluție unică vectorul $Z = \{Z_1, \ldots . Z_m)$ *ale cărui componente sînt limbaje regulate.*

Demonstrație. Vom proceda prin inducție asupra lui m (dimensiunea matricii E). Pentru $m = 1$, soluția este $Z = BE^*$ (v. consecința I.6.1). Conform teoremei II.2.3, E^* este un limbaj regulat și din aceeași teoremă rezultă că și Z este un limbaj regulat.

Să presupunem că teorema este valabilă pentru $m = \overline{1, n-1}$ și fie sistemul :

$Z = ZE \cup B$, E fiind o matrice de tip (n, n). Cum $e \notin E_{nn}$ (în caz contrar matricea ar fi singulară) avem :

(II.4.1) $Z_n = (Z_1 E_{1n} \cup \ldots \cup Z_{n-1} E_{n-1,n} \cup B_n) E_{nn}^*$, soluția unică a ecuației :

$$Z_n = Z_n E_{nn} \cup (Z_1 E_{1n} \cup \ldots \cup Z_{n-1} E_{n-1,n} \cup B_n),$$

(ultima ecuație a sistemului considerat). Eliminînd astfel pe Z_n din primele $n-1$ ecuații, obținem astfel sistemul :

(II.4.2) $Z_j = Z_1 (E_{1j} \cup E_{1n} E_{nn}^* E_{nj}) \cup \ldots \cup Z_{n-1} (E_{n-1,j} \cup E_{n-1n} E_{nn}^* E_{nj}) \cup B_n E_{nn}^* E_{nj} \cup B_j$, $j = \overline{1, n-1}$, în care matricea coeficienților este $E' = (E_{kj} \cup E_{kn} E_{nn}^* E_{nj})$ o matrice de tip $(n-1, n-1)$, avînd drept elemente limbaje regulate. Matricea E' este nesingulară. Într-adevăr, dacă $e \in E'_{kj}$, atunci sau $e \in E_{kj}$ sau $e \in E_{kn}$ și $e \in E_{nj}$ deci existența unui monom strict ciclic $E'_{j_1 j_2} \ldots E'_{j_{t-1} j_1}$, care conține cuvîntul e ar antrena existența unui monom strict ciclic în matricea E, care ar conține cuvîntul e, ceea ce este imposibil, E fiind o matrice nesingulară.

Prin urmare, sistemul (II.4.2) admite o soluție unică si din ipoteza inductivă componentele soluției sînt limbaje regulate. Din relația (II.4.1) rezultă că și Z_n este un limbaj regulat. ■

Consecința II.4.1. Demonstrarea teoremei precedente oferă și un algoritm de găsire a soluției, algoritm care constă din eliminări succesive.

Exemplul II.4.2. Fie alfabetul $\{e, 1\}$ și sistemul :

$$(Z_1 Z_2 Z_3) = (Z_1 Z_2 Z_3) \begin{pmatrix} \{0\} & \{1, 1\} & \{1\} \\ \{e\} & \{e\} & \{e\} \\ \{e\} & \{e\} & \varnothing \end{pmatrix} \cup (\{e\}, \{1\}^*, \varnothing)$$

care pe componente revine la :

$$Z_1 = Z_1 0 \cup Z_2 \cup Z_3 \cup \{e\}$$
$$Z_2 = Z_1 11 \cup Z_2 0 \cup Z_3 0 \cup \{1\}^*$$
$$Z_3 = Z_1 1 \cup Z_2 .$$

Substituindu-l pe Z_3 în primele două ecuații, obținem :

$$Z_1 = Z_1 \{0, 1\} \cup Z_2 \cup \{e\}$$

$$Z_2 = Z_1 \{11, 10\} \cup Z_2 0 \cup \{1\}^*.$$

Rezolvînd a doua ecuație, găsim :

$$Z_2 = (Z_1 \{11, 10\} \cup 1^*) 0^*,$$

care introdusă în prima ecuație ne dă :

$$Z_1 = Z_1 (\{0, 1\} \cup \{11, 10\} \{0\}^* \cup \{1\}^* \{0\}^*),$$

adică :

$$Z_1 = \{1\}^* \{0\}^* (\{0, 1\} \cup \{11, 10\} \{0\}^*)^* = \{0, 1\}^*,$$

de unde obținem imediat pe Z_2 și Z_3.

§II.5. LIMBAJE DE TIP 2 (LIMBAJE LIBERE DE CONTEXT)

Limbajele de care ne vom ocupa în acest paragraf modelează proprietățile limbajelor de programare.

Teorema II.5.1. *Fie $G = (I_N, I_T, x_0; F)$ o gramatică de tipul 2. Există atunci o gramatică $G_1 = (I_N, I_T, x_0; F_1)$, de tipul 2, astfel încît $L(G_1) = L(G) \setminus \{e\}$ și G_1 nu are reguli de generare de forma $x \to e$.*

Demonstrație. Să construim mulțimea F_1. Pentru aceasta fie $I_1 = \{x \mid x \in I_N, (x \to e) \in F\}$ și $I_{j+1} = I_j \cup \{x \mid x \in I_N, (x \to p) \in F, p \in I_j^*\}$, $j \geqslant 1$. Există $k \in N$ astfel încît $I_k = I_{k+1}$, deoarece mulțimea F este finită. Mai mult, $I_k = I_{k+v}$, $\forall v \in N$.

Avem deci $x \overset{*}{\Rightarrow} e$, dacă și numai dacă $x \in I_k$, așa cum rezultă din modul de formare al mulțimilor I_j.

Alegem $F_1 = \{(x \to p_1), p_1 \neq e, \exists (x \to p) \in F, p_1$ se obține din p prin îndepărtarea simbolilor din $I_k\}$. Vom arăta că $L(G_1) = L(G) \setminus \{e\}$.

Fie $p \in L(G_1)$. Avem deducția :

$$x_0 \underset{G_1}{\Rightarrow} p_1 \underset{G_1}{\Rightarrow} p_2 \underset{G_1}{\Rightarrow} \ldots \underset{G_1}{\Rightarrow} p \in I_T^*,$$

sau

$$x_0 \underset{G_1}{\Rightarrow} p_1' x p_1'' \underset{G_1}{\Rightarrow} p_1' r p_1'' \underset{G_1}{\overset{*}{\Rightarrow}} p,$$

ceea ce atrage existența regulilor de generare de forma $(x_1 \to r_1) \in F_1$. Corespunzător, în gramatica G avem reguli de forma $x_1 \to r$, cuvîntul r conținînd pe lîngă simbolii din cuvîntul r_1, și simboli neterminali din I_k (deci pentru care avem reguli de generare de forma $x' \to e$, care nu alterează rezultatul final al deducției).

În gramatica G avem o deducție de forma:

$$x_0 \underset{G}{\Rightarrow} p_1' x_1 p_1'' \underset{G}{\Rightarrow} p_1' r p_1'' \underset{G}{\Rightarrow} \ldots \underset{G}{\Rightarrow} p \in I_T^*$$

și $p \in L(G) \setminus \{e\}$.

Invers, să arătăm că $L(G) \setminus \{e\} \subseteq L(G_1)$. Pentru aceasta să considerăm gramatica $G_2 = (I_N, I_T, x_0 ; F \cup F_1)$, pentru care avem evident $L(G) \subseteq L(G_2)$ și deci $L(G) \setminus \{e\} \subseteq L(G_2) \setminus \{e\}$.

Rămîne să arătăm că $L(G_2) \setminus \{e\} \subseteq L(G_1)$. Fie $p \in L(G_2) \setminus \{e\}$. Atunci în deducția $x_0 \underset{G_2}{\Rightarrow} p$ pot apare reguli de generare din F și din F_1. Pornind de la această deducție, putem construi o altă deducție a cuvîntului p, folosind exclusiv reguli de generare din F_1, înlocuind regulile de generare din F cu regulile corespunzătoare din F_1 (obținute din regulile de forma $(x \to p) \in F$ prin îndepărtarea din p a simbolilor neterminali din I_k). ■

Teorema II.5.2. *Fie $G = (I_N, I_T, x_0 ; F)$ o gramatică de tip 2. Dacă $e \in L(G)$, atunci există o gramatică $G' = (I'_N, I_T, x_0 ; F')$ de tipul 2, astfel încît $L(G) = L(G')$ și singura regulă de generare de forma $x \to e$ este regula $x_0' \to e$ și x_0' nu apare în partea dreaptă a unei reguli de generare din F'.*

Demonstrație. Fie G_1 gramatica construită în demonstrarea teoremei precedente și G' gramatica de tip 2:
$G' = (I_N \cup \{x_0'\}, I_T, x_0' ; F_1 \cup \{x_0' \to e, x_0' \to x_0\})$. Gramatica G' satisface condițiile cerute în teoremă și $L(G') = L(G_1) \cup \{e\} = L(G)$. ■

Folosind noțiuni din teoria grafurilor, vom prezenta un mod de descriere a deducțiilor din gramaticile de tip 2.

Fie $G = (I_N, I_T, x_0 ; F)$ o gramatică de tip 2.

Definiția II.5.1. *Un arbore de deducție* în gramatica G este un arbore care se bucură de următoarele proprietăți:

1. — orice nod este etichetat cu un simbol din $I_N \cup I_T$;
2. — rădăcina arborelui este etichetată cu x_0;
3. — nodurile care au cel puțin un descendent propriu, sînt etichetate cu simboli din I_N;
4. — dacă $v_1, v_2, \ldots, v_k$ sînt descendenți ai nodului v, scriși într-o anume ordine introdusă pe mulțimea descendenților nodului v, atunci în gramatica G există regula de generare $x \to y_1 y_2 \ldots y_k$, unde x este eticheta nodului v, iar y_j este eticheta nodului v_j, $j = 1, k$;
5. — etichetele nodurilor terminale ale arborelui (în ordinea în care apar) descriu cuvîntul generat prin deducția reprezentată de arbore.

Exemplul II.5.1. Fie gramatica $G = (\{x_0, x\}, \{i_1, i_2\}, x_0 ; F)$, unde $F = \{x_0 \to i_1 x x_0,\ x_0 \to i_1,\ x \to x_0 i_2 x,\ x \to i_2 i_1,\ x \to x_0 x_0\}$.

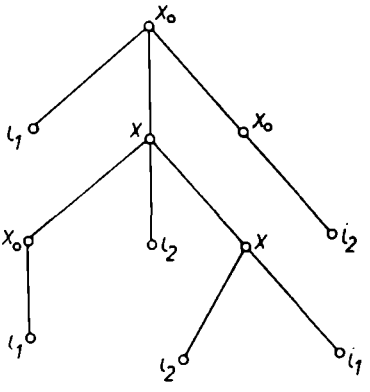

Fig. II. 5.1

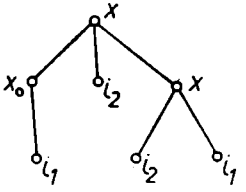

Fig. II. 5.2

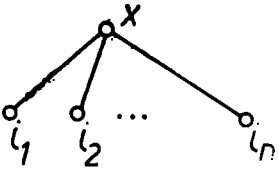

Fig. II. 5.3

Arborele din figura II.5.1 corespunde deducției : $x_0 \Rightarrow i_1 x x_0 \Rightarrow i_1 x_0 i_2 x x_0 \Rightarrow$ $\Rightarrow i_1 x_0 i_2 x i_1 \Rightarrow i_1 i_1 i_2 x i_2 \Rightarrow i_1 i_1 i_2 i_2 i_1 i_1$. ■

Definiția II.5.2. *Un subarbore* al unui arbore de deducție este subgraful generat de un nod și mulțimea descendenților săi, păstrîndu-se etichetele din arbore. Eticheta rădăcinii unui subarbore poate fi alta decît eticheta nodului inițial.

Exemplul II.5.2. Subgraful din figura II.5.2 este un subarbore al arborelui din figura II.5.1. Acestui subgraf îi corespunde deducția : $x \Rightarrow x_0 i_2 x \Rightarrow$ $\Rightarrow i_1 i_2 x \Rightarrow i_1 i_2 i_2 i_1$.

Teorema II.5.3. *Fie* $G = (I_N, I_T, x_0; F)$ *o gramatică de tip 2. Atunci* $p \in L(G) \setminus \{e\}$, *dacă și numai dacă există un arbore de deducție corespunzător deducției* $x_0 \overset{*}{\underset{G}{\Rightarrow}} p$.

Demonstrație. Pentru $x \in I_N$ să considerăm gramatica $G_x = (I_N, I_T, x; F)$. Oricare ar fi $x \in I_N$ regulile de generare sînt aceleași pentru toate gramaticile G_x, deci $x \overset{*}{\underset{G'_x}{\Rightarrow}} p$ echivalează cu $x \overset{*}{\underset{G''_x}{\Rightarrow}} p$, $\forall x', x'' \in I_N$ și cum $G_{x_0} = G$, aceasta revine la $x \overset{*}{\underset{G}{\Rightarrow}} p$.

Să arătăm mai întîi că dacă există un arbore de deducție în gramatica G_x, ale cărui noduri terminale descriu cuvîntul p, atunci există deducția $x \overset{*}{\underset{G_x}{\Rightarrow}} p$. Demonstrația se va face prin inducția după numărul n, al nodurilor neterminale din arbore.

Fie $n = 1$. În acest caz arborele are aspectul din figura II.5.3. În acest caz : $p = i_1 i_2 ..., i_n$ și $(x \to p) \in F$.

Să presupunem că afirmația este valabilă pentru arborii cu cel mult $n-1$ noduri neterminale și că p este descris de un arbore de deducție a cărui rădăcină este etichetată cu x și care are n noduri neterminale, $n > 1$.

Să considerăm nodurile etichetate cu simbolii $y_1 \ldots y_m \in I_N \cup I_T$, descendenți direcți ai rădăcinii x, care nu pot fi toate noduri terminale (în caz contrar arborele care descrie cuvîntul p ar avea un singur nod neterminal). Avem, evident, $(x \to y_1 \ldots y_m) \in F$. Fie $\{y_{j_1} \ldots y_{j_r}\} = \{y_1 \ldots y_m\} \cap I_N$. Nodurile $(y_{j_h} \mid h = \overline{1, r})$ constituie rădăcinile unor subarbori, care au cel mult $n-1$ vîrfuri neterminale. Fie $q_1, \ldots, q_r$ cuvintele descrise de acești subarbori.

În baza ipotezei inductive, avem în gramaticile $G_{y_{j_1}} \ldots G_{y_{j_r}}$ deducțiile $y_{j_h} \overset{*}{\underset{G_{y_{j_h}}}{\Rightarrow}} q_h$, $h = \overline{1, r}$. Este clar că dacă $h \leqslant k$, descendenții nodului y_{j_h} (care descriu cuvîntul q_h) apar la stînga descendenților nodului y_{j_k} (care descriu pe q_k). Ca atare, avem în gramatica G_x deducția $x \overset{*}{\underset{G_x}{\Rightarrow}} y_1 \ldots y_m = y_1 \ldots$ $\ldots y_{j_1} \ldots y_{j_r} \ldots y_m \Rightarrow i_1 \ldots q_1 \ldots q_r \ldots i_n = p$, unde $\{i_1, \ldots, i_r\} = \{y_1 \ldots y_m\} \cap I_T$.

Reciproc, să presupunem că în gramatica G_x există deducția $x \overset{*}{\Rightarrow} p$. Vom arăta că există un arbore de deducție în gramatica G_x care descrie cuvîntul p. Vom proceda inductiv după lungimea n a deducției de mai sus. Dacă $n = 1$, atunci $(x \to p) \in F$ și există un arbore al cărui aspect este redat în figura II.5.3.

Să presupunem că afirmația este adevărată pentru $n-1$ și că deducția $x \overset{*}{\underset{G_x}{\Rightarrow}} p$ are lungimea n. Fie $x \to y_1 \ldots y_m$ prima regulă de generare utilizată. Este clar că orice simbol din p trebuie, fie să aparțină mulțimii $\{y_1, \ldots, y_m\} \cap I_T$, fie să poată fi dedus dintr-un simbol din mulțimea $\{y_1, \ldots, y_m\} \cap I_N$. Astfel q_j, subcuvîntul din p, care se deduce din y_j, se află la stînga subcuvîntului q_k, care se deduce din y_k pentru $j < k$.

Astfel putem scrie $p = p_1 p_2 \ldots, p_n$, unde $p_j \in I_T \cup \{q_1, \ldots, q_m\}$ și $y_j \overset{*}{\underset{G_{y_j}}{\Rightarrow}} p_j$, $\forall j = \overline{1, m}$. Prin ipoteza inductivă există un arbore de deducție în gramatica G_{y_j} care descrie subcuvîntul p_j. Arborele obținut prin conectarea rădăcinilor acestor arbori la nodul x va constitui arborele de deducție care descrie cuvîntul p. ■

Noțiunea de arbore de deducție poate fi utilizată în demonstrarea următoarei teoreme:

Teorema II.5.4. *Este decidabil faptul că limbajul general de o gramatică de tip 2 este vid.*

Demonstrație. Vom construi un algoritm prin care să se determine dacă un limbaj generat de o gramatică de tip 2 este nevid. Fie $G = (I_N, I_T, x_0; F)$ o gramatică de tip 2. Să considerăm deducția $x_0 \overset{*}{\Rightarrow} p$ și arborele corespunzător. Vom atașa acestui arbore un arbore de deducție dintr-o mulțime finită

de arbori. Să presupunem că există un drum în arbore cu două noduri v_1 și v_2 etichetate cu același simbol y situate la înălțimile n_1 și n_2 respectiv, cu $n_2 > n_1$. Subarborelui cu rădăcina în v_1 îi corespunde o deducție $y \overset{*}{\Rightarrow} p_1$, iar celui cu rădăcina în v_2 deducția $y \overset{*}{\Rightarrow} p_2$. Să observăm că p_2 trebuie să fie un subcuvînt al lui p_1, eventual chiar p_1. Cuvîntul p se poate scrie sub forma $p = p_3 p_1 p_4$, unde p_3 sau p_4 pot fi eventual cuvîntul nul. Dacă înlocuim subarborele cu rădăcina în v_1 prin acela cu rădăcina în v_2 obținem un cuvînt $p_3 p_2 p_4$ (eventual chiar cuvîntul p) și $x_0 \overset{*}{\Rightarrow} p_3 p_2 p_4$. În acest fel am eliminat cel puțin un nod v din arbore. Dacă noul arbore are un drum cu două noduri etichetate la fel, aplicăm același procedeu cuvîntului $p_3 p_2 p_4$. Procedeul se poate aplica de un număr finit de ori, deoarece fiecare pas elimină cel puțin un nod din arbore și arborele are un număr finit de noduri. Arborele rămas după epuizarea aplicării acestui procedeu conține în fiecare drum noduri etichetate cu simboluri dinstincte, deci are drumurile de lungime cel mult egale cu $|I_N \cup I_T|$. Conchidem că dacă $L(G) \neq \varnothing$, există o deducție $x_0 \overset{*}{\underset{G}{\Rightarrow}} p$, $p \in L(G)$ căreia îi corespunde un arbore de deducție avînd drumurile de lungime cel mult $|I_N \cup I_T|$. Cum mulțimea arborilor de deducție ale căror drumuri au lungimea cel mult $|I_N \cup I_T|$ este finită, putem afirma că dacă $L(G) \neq \varnothing$ există un arbore de deducție aparținînd acestei mulțimi finite și astfel într-un număr finit de pași se poate deduce valabilitatea propoziției „$L(G) \neq \varnothing$". ■

Vom studia în continuare unele proprietăți ale limbajelor de tip 2, legate de așa-numitele forme normale.

Teorema II.5.5. *Fie $G = (I_N, I_T, x_0 ; F)$ o gramatică de tip 2 cu $L(G) \neq \varnothing$. Există o gramatică echivalentă G_1, astfel că pentru orice $x \in I_N$ să existe o deducție $x \overset{*}{\underset{G_1}{\Rightarrow}} p$.*

Demonstrație. Atașăm fiecărui simbol $x \in I_N$ gramatica $G_x = (I_N, I_T, x ; F)$. Dacă $L(G_x) = \varnothing$, atunci putem elimina x din I_N și toate regulile de generare din F de forma $x \to q$ sau $x_1 \to \ldots x \ldots$, fără a modifica limbajul generat de gramatica G. Anume, fie G_1 gramatica astfel obținută din G, $G_1 = (I'_N, I_T, x_0 ; F')$.

Să arătăm că $L(G_1) = L(G)$. Este clar că $L(G_1) \subseteq L(G)$, deoarece orice deducție în G_1 este o deducție în G.

Invers, să arătăm că $L(G) \subseteq L(G_1)$. Vom proceda prin reducere la absurd. Să presupunem că există $p \in L(G)$ și $p \notin L(G_1)$. Atunci există o deducție de forma $x_0 \Rightarrow p_1 x p_2 \overset{*}{\Rightarrow} p \in L(G)$, $x \in I_N \setminus I'_N$.

Astfel trebuie să existe o deducție de forma $x \overset{*}{\underset{G}{\Rightarrow}} q \in L(G_x)$, ceea ce contrazice faptul că $L(G_x) = \varnothing$. ■

Observația II.5.1. Teorema precedentă ne permite să eliminăm din alfabetul inițial al gramaticii toate simbolurile care nu apar efectiv în deducție.

Consecința II.5.1. Fie G o gramatică de tip 2 cu $L(G) \Rightarrow \varnothing$. Atunci există o gramatică $G' = (I'_N, I_T, x_0; F')$ de tip 2, cu $L(G') = L(G)$ astfel încît $\forall x \in I_N$, există o deducție de forma $x_0 \overset{*}{\Rightarrow} p_1 x p_3 \Rightarrow p_1 p_2 p_3$, $p_1, p_2, p_3 \in I_T^*$.

Demonstrație. Fie $G_1 = (I_N, I_T, x_0; F)$ o gramatică care generează pe $L(G)$ și satisface condițiile din teorema precedentă. Dacă $x_0 \overset{*}{\underset{G_1}{\Rightarrow}} q_1 x q_2$, $q_1, q_2 \in (I_N \cup I_T)^*$, atunci există o deducție $x_0 \overset{*}{\underset{G_1}{\Rightarrow}} p_1 x p_3 \overset{*}{\underset{G_1}{\Rightarrow}} p_1 p_2 p_3 \in I_T^*$, deoarece pentru $\forall y \in I_N$ există o deducție $y \overset{*}{\underset{G_1}{\Rightarrow}} p \in I_T^*$. Fie I'_N mulțimea simbolurilor x cu proprietatea că există o deducție $x_0 \overset{*}{\underset{G_1}{\Rightarrow}} q_1 x q_2$. Această mulțime o putem construi efectiv. Astfel, $x_0 \in I'_N$; apoi adăugăm mulțimii orice simbol care apare în partea dreaptă a unei reguli de generare $x \to q$ dacă x aparține deja mulțimii, procedeu finitar, deoarece numărul de reguli de generare este finit.

Mulțimea F' se obține prin eliminarea regulilor de generare care conțin simboli din $I_N \setminus I'_N$. Este clar că $L(G') = L(G)$. ■

Definiția II.5.3. O *deducție imediată la stînga* a cuvîntului $p \in L(G)$, G fiind o gramatică de tip 2, este o deducție de forma:
$x \Rightarrow q_1 \Rightarrow q_2 \Rightarrow \ldots q_n \Rightarrow p$, unde $q_j = r_k x_j u_j$, $r_j \in I_T^*$, $u_j \in (I_N \cup I_T)^*$ și $(x_j \to v_j) \in F$, $v_j \in (I_N \cup I_T)^*$, iar $q_{j+1} = r_j v_j u_j$.

În mod analog se introduc *deducțiile imediate la dreapta.*

Observația II.5.2. Unei deducții imediate la stînga (la dreapta) îi corespunde în mod biunivoc un arbore de deducție.

Într-adevăr, plecînd de la un arbore de deducție obținem o deducție imediată la stînga (la dreapta), aplicînd sistematic substituirea cu prioritate a primului (ultimului) simbol care apare în descendenții unui nod oarecare. Reciproca este evidentă.

Teorema II.5.6. *Într-o gramatică de tip 2 pentru orice cuvînt $p \in L(G)$ există o deducție imediată la stînga.*

Demonstrație. Vom proceda prin inducție asupra lungimii n a deducției, arătînd că pentru orice deducție $x_0 \overset{*}{\Rightarrow} p$ există o deducție imediată la stînga prin care se obține cuvîntul p.

Dacă $n = 1$, avem $(x_0 \to p) \in F$ și deci deducția imediată la stînga $x_0 \Rightarrow p$.

Să presupunem afirmația adevărată pentru $n \leqslant k$ și fie $x_0 \overset{*}{\Rightarrow} q_1 \Rightarrow p$ o deducție de lungime $k+1$ cu $q_1 = y_1 \ldots y_m$, $y_j \in I_N \cup I_T$, $j = \overline{1, m}$. Putem scrie $p = r_1 r_2 \ldots r_m$, unde $y_j \overset{*}{\Rightarrow} r_j$, $j = \overline{1, m}$.
Prin ipoteza inductivă există deducții imediate la stînga de la y_j la r_j, $j = \overline{1, n}$. Dacă $y_j \in I_T$, atunci $r_j = y_j$.

Dacă $y_1 \in I_N$ avem deducția imediată la stînga $x_0 \overset{*}{\Rightarrow} y_1 \dots y_m \Rightarrow r_1 y_2 \dots y_m$. În continuare vom utiliza pentru deducerea cuvîntului $r_1 r_2 y_3 \dots y_m$ deducția imediată la stînga a cuvîntului r_2 din y_2, obținînd astfel deducția la stînga :

$$x_0 \overset{*}{\Rightarrow} y_1 \dots y_m \overset{*}{\Rightarrow} r_1 y_2 \dots y_m \Rightarrow r_1 r_2 \dots y_m \text{ ș. a. m. d. } \blacksquare$$

Definiția II.5.4. O *gramatică G de tip 2* este *ambiguă* dacă există un cuvînt $p \in L(G)$ pentru care avem cel puțin două deducții imediate la stînga distincte, pornind de la același simbol neterminal.

În caz contrar G este *neambiguă*.

Exemplul II.5.3. Fie gramatica $G = (\{x_0, x\}, \{i_1, i_2\}, x_0 ; \{x_0 \to i_1, x_0 \to i_1 i_2 x_0 i_2, x_0 \to i_1 x i_2, x \to i_2 x_0 ; x \to i_1 x x i_2)$. Atunci pentru cuvîntul $i_1 i_2 i_1 i_2 \in L(G)$ există următoarele două deducții imediate la stînga :

$$x_0 \Rightarrow i_1 i_2 x_0 i_2 \Rightarrow i_1 i_2 i_1 i_2 ,$$

și

$$x_0 \Rightarrow i_1 x i_2 \Rightarrow i_1 i_2 x_0 i_2 \Rightarrow i_1 i_2 i_1 i_2 ,$$

deci gramatica G este ambiguă.

Observația II.5.3. Gramaticile neambigue sînt importante în special pentru procesele de compilare a programelor scrise în limbaje de programare.

Exemplul II.5.4. Fie $G_1 = (\{x_0\}, \{i\}, x_0 ; \{x_0 \to x_0 x_0, x_0 \to i^5\})$ și $G_2 = (\{y_0\}, \{i\}, y_0 ; \{y_0 \to i^5, y_0 \to i^5 y_0\})$.
Atunci $L(G_1) = L(G_2) = \{i^{5k} \mid k \geqslant 1\}$. Gramatica G_1 este ambiguă deoarece pentru cuvîntul i^{15} avem deducțiile imediate la stînga :

$$x_0 \Rightarrow x_0 x_0 \Rightarrow x_0 x_0 x_0 \Rightarrow i^5 x_0 x_0 \Rightarrow i^{10} x_0 \Rightarrow i^{15},$$

și

$$x_0 \Rightarrow x_0 x_0 \Rightarrow i^5 x_0 \Rightarrow i^5 x_0 x_0 \Rightarrow i^{10} x_0 \Rightarrow i^{15},$$

în timp ce gramatica G_2 este neambiguă.

Acest exemplu arată că, pentru același limbaj pot exista atît gramatici ambigue cît și neambigue care să-l genereze. De aceea are sens definiția :

Definiția II.5.5. Un *limbaj* este *inerent ambiguu* dacă toate gramaticile care-l generează sînt ambigue. În caz contrar (adică în cazul existenței unei gramatici neambigue care generează limbajul) acesta se zice *neambiguu*.

Teorema II.5.7. *Dacă* L_j, $j = \overline{1, k}$ *sînt limbaje neambigue disjuncte două cîte două, atunci limbajul* $\bigcup_{j=1}^{k} L_j$ *este un limbaj neambiguu.*

Demonstrație. Vom presupune pentru simplitate $k = 2$, demonstrația pentru $k > 2$ făcîndu-se la fel.

Fie $G_j = (I_{Nj}, I_T, x_{0j}; F_j)$, $L_j = L(G_j)$, $j = 1, 2$ două gramatici neambigue cu același alfabet terminal, $L_1 \cap L_2 = \varnothing$. Putem presupune că $(I_{N1} \setminus I_T) \cap (I_{N2} \setminus I_T) = \varnothing$. Fie $G = (I_N, I_T, x_0; F)$, unde $x_0 \in I_{N1} \cup I_{N2}$. $I_N = I_{N1} \cup I_{N2} \cup \{x_0\}$, $F = F_1 \cup F_2 \cup \{x_0 \to x_{01}, x_0 \to x_{02}\}$. Avem $L(G) = L_1 \cup L_2$. Fie $x_0 \underset{G}{\Rightarrow} p_1 \underset{G}{\Rightarrow} \dots \underset{G}{\Rightarrow} p_s$ și $x_0 \underset{G}{\Rightarrow} q_1 \dots \underset{G}{\Rightarrow} q_t$ două deducții imediate la stînga ale cuvîntului $p_s = q_t$ în gramatica G.

Deoarece în F sînt numai două reguli de generare de forma $x_0 \to z$, și anume $x_0 \to x_{01}$ și $x_0 \to x_{02}$, urmează că $p_1, q_1 \in \{x_{01}, x_{02}\}$.

Să presupunem că $p_1 = q_1 = x_{01}$. Deoarece $(I_{N1} \setminus I_T) \cap (I_{N2} \setminus I_T) = \varnothing$, aici regulile de generare folosite în ambele deducții provin din F_1 și cum G_1 era neambiguă, cele două deducții coincid. În mod analog deducțiile coincid dacă $p_1 = q_1 = x_{02}$.

Să presupunem că $p_1 = x_{01}$ și $q_1 = x_{02}$. Atunci deducția $x_0 \overset{\bullet}{\underset{G}{\Rightarrow}} p_s$ include numai reguli din F_1, deci $p_s \in L_1$, iar $x_0 \overset{\bullet}{\underset{G}{\Rightarrow}} q_t$ include numai reguli din F_2, deci $q_t \in L_2$.

Prin urmare cuvîntul $p_s = q_t \in L_1 \cap L_2 = \varnothing$, ceea ce este imposibil. Deci cazul $p = x_{01}$, $q_1 = x_{02}$ nu poate apare și gramatica G este neambiguă. ■

Teorema II.5.8. *Pentru orice gramatică G de tipul 2 există o gramatică echivalentă G_1 care nu conține reguli de generare de tipul $x_1 \to x_2$, $x_1, x_2 \in I_N$.*

Demonstrație. Fie $G = (I_N, I_T, x_0; F)$ și $F = F' \cup F''$, unde F' este mulțimea regulilor de generare de forma $x_1 \to x_2$, $x_1, x_2 \in I_N$ și $F'' = F \setminus F'$.

Construim gramatica $G_1 = (I_N, I_T, x_0; F_1)$ unde F_1 se obține din F prin înlocuirea regulilor de generare din F în modul următor:
dacă $(x_1 \to x_2) \in F_1$ și $\{(x_2 \to \omega_j) \mid \omega_j \notin I_N\} \subseteq F$ înlocuim $(x_1 \to x_2)$ cu mulțimea de reguli $\{(x_1 \to \omega_j)\}$.

Să arătăm că $L(G_1) = L(G)$. Se vede imediat că $L(G_1) \subseteq L(G)$.

Fie $p \in L(G)$. Avem $x_0 \overset{\bullet}{\underset{G}{\Rightarrow}} p$. Conform teoremei II.5.6 există o deducție imediată la stînga:
$x_0 \Rightarrow q_1 \Rightarrow q_2 \Rightarrow \dots \Rightarrow q_m = p$, unde $q_j = r_j x_j u_j$, $q_{j+1} = r_j v_j u_j$, $r_j \in I_T^*$, $u_j, v_j \in (I_N \cup I_T)^*$ și $(x_j \to v_j) \in F$. Dacă $(x_j \to v_j) \in F''$, $j = \overline{1, m}$, atunci $(x_j \to v_j) \in F_1$ și $p \in L(G_1)$. Fie k cel mai mic indice pentru care $(x_k \to v_k) \in F'$ și presupunem că $(x_j \to v_j) \in F'$ pentru $k \leqslant j < h$, iar $(x_h \to v_h) \in F''$. Atunci cuvintele $q_k, q_{k+1}, \dots, q_{h-1}$ au toate aceeași lungime și cum deducțiile sînt imediate la stînga, simbolii înlocuiți ocupă aceleași poziții, adică deducția $q_k \Rightarrow q_{k+1} \Rightarrow \dots \Rightarrow q_{h-1}$ are de fapt aspectul $r_k v_k u_k \Rightarrow r_k x_{k+1} u_k \Rightarrow \dots \Rightarrow r_k x_{h-1} u_k$, ceea ce implică $q_k \Rightarrow r_k v_h u_k = q_h$, deducție obținută prin utilizarea unei reguli din F_1. Astfel $p \in L(G_1)$ și $L(G) = L(G_1)$. ■

Teorema II.5.9. (*teorema lui Noam Chomsky de existență a formei normale*). *Orice gramatică G de tip 2 cu $e \notin L(G)$ este echivalentă cu o gramatică G_1 de*

tip 2, în care regulile de generare sînt de forma $x \to x_1 x_2$ *sau* $x \to i$, $x\, x_1, x_2 \in I_N$, $i \in I_T$.

Demonstrație. Fie G o gramatică de tip 2. Conform teoremei II.5.8 există o gramatică echivalentă $G_1 = (I_N, I_T, x_0; F)$ de același tip, care nu conține reguli de generare de fomra $x_1 \to x_2 x_1$, $x_2 \in I_N$.

Astfel, dacă $(x \to y) \in F$, atunci $y \in I_T$ și regula de generare are forma $x \to i$, admisă în teoremă.

Fie $(x \to y_1 y_2 \dots y_m) \in F$, $m \geqslant 2$, $\{y_{j_1}, \dots, y_{j_k}\} = \{y_1, \dots, y_m\} \cap I_T$ și $\{z_1, \dots z_k\}$ o mulțime de k simboluri. Să considerăm o nouă mulțime de reguli de generare F' obținută prin adăugarea la F a regulilor de forma $z_h \to y_{j_h}$ (care sînt de forma prescrisă) și înlocuirea regulii $x \to y_1 y_2 \dots y_m$ prin $x \to v_1 \dots v_m$, unde:

$$v_j = \begin{cases} y_j, & \text{dacă } y_j \in I_N \\ z_j, & \text{dacă } y_j \in I_T \end{cases}$$

Gramatica $G_2 = (I'_N, I_T, x_0 : F')$, unde I'_N se obține prin adăugarea la I_N a tuturor simbolilor z_h, este echivalentă cu gramatica G_1, deci cu gramatica G.

Într-adevăr, dacă $p \overset{*}{\underset{G_1}{\Rightarrow}} q$, atunci $p \overset{*}{\underset{G_2}{\Rightarrow}} q$, deci $L(G_1) \subseteq L(G_2)$. Vom arăta prin inducție relativ la lungimea unei deducții că dacă $x \overset{*}{\underset{G_2}{\Rightarrow}} p$, unde $x \in I_N \subseteq I'_N$, $p \in I_T^*$, atunci $x \overset{*}{\underset{G_1}{\Rightarrow}} p$.

Afirmația este evidentă pentru deducții de lungime 1.

S-o presupunem valabilă pentru deducții de lungime cel mult k și fie $x \overset{*}{\underset{G_2}{\Rightarrow}} p$ o deducție de lungime $k+1$. În mod necesar, prima regulă de generare utilizată este de forma $x \to v_1 \dots v_m$, $m \geqslant 2$. Putem scrie $p = p_1 \dots p_m$, unde $v_j \overset{*}{\underset{G_2}{\Rightarrow}} p_j$, $j = \overline{1, m}$, este o deducție de lungime cel mult k. Dacă $v_j \in I'_N \setminus I_N$, atunci există o singură regulă de generare din F', pe care o putem folosi și anume $v_j \to i_j$, $i_j \in I_T$. În acest caz $i_j = p_j$.

Din modul de a-l defini pe F', rezultă existența unei reguli de generare în F de forma $x \to y_1 \dots y_m$, unde

$$y_j = \begin{cases} v_j, & \text{dacă } v_j \in I_N \\ i_j, & \text{dacă } v_j \in I'_N \setminus I_N. \end{cases}$$

Pentru $v_j \in I_N$ deducția $v_j \overset{*}{\underset{G_2}{\Rightarrow}} p_j$ are lungimea cel mult k și prin ipoteza inductivă, avem $y_j \overset{*}{\underset{G_1}{\Rightarrow}} p_j$. Prin urmare $x \overset{*}{\underset{G_1}{\Rightarrow}} p$ și $L(G_2) \subseteq L(G_1)$.

Să observăm că G_2 nu are forma impusă de enunț, deoarece conține reguli de generare de forma $x \to t_1 t_2 \ldots t_m$, $m \geqslant 2$, $t_j \in I_N$.

Construim o nouă gramatică $G_3 = (I_N'', I_T, x_0; F'')$ astfel: pentru fiecare regulă de generare $(x \to t_1 t_2 \ldots t_m) \in F'$, $m \geqslant 3$ introducem noi variabile $u_1, u_2, \ldots, u_{m-2}$ și înlocuim $x \to t_1 t_2 \ldots t_m$ prin mulțimea de reguli de generare $\{x \to t_1 u_1, u_1 \to t_2 u_2, \ldots, u_{m-2} \to t_{m-2} u_{m-2}, u_{m-2} \to t_{m-1} t_m\}$; I_N'' se obține prin adăugarea la I_N' a simbolilor u_j, iar F'' se obține din F' prin înlocuirea de mai sus.

Avem $L(G_2) \subseteq L(G_3)$, deoarece, dacă $x \overset{*}{\underset{G_2}{\Rightarrow}} p$ atunci $x \overset{*}{\underset{G_3}{\Rightarrow}} p$. Invers, să arătăm că $L(G_3) \subseteq L(G_2)$. Fie $p \in L(G_3)$. Vom proceda prin inducție asupra lungimii l a deducției $x \overset{*}{\underset{G_3}{\Rightarrow}} p$.

Pentru cazurile $l = 1$ și $l = 2$ avem evident $p \in L(G_2)$. Să presupunem că pentru toate deducțiile $x \overset{*}{\underset{G_3}{\Rightarrow}} p$, de lungime cel mult k există deducția $y \overset{*}{\underset{G_2}{\Rightarrow}} p$, unde y coincide cu x dacă $x \in I_N'$ și fie $x \overset{*}{\underset{G_3}{\Rightarrow}} p$ o deducție cu lungime $k+1$. Prima regulă de generare din această deducție este obligatoriu de forma $x \to w_1 w_2$, unde $w_1, w_2 \in I_N''$.

Dacă $w_1, w_2 \in I_N'$ regula de generare $x \to w_1 w_2$ aparține lui F', din gramatica G_2 și deci există deducția $x \Rightarrow w_1 w_2 \overset{*}{\underset{G_3}{\Rightarrow}} p = p_1 p_2$, unde $w_1 \overset{*}{\underset{G_3}{\Rightarrow}} p_1$, $w_2 \overset{*}{\underset{G_3}{\Rightarrow}} p_2$ sînt deducții de lungime cel mult k.

Aplicînd ipoteza inductivă, există deducțiile $w_1 \overset{*}{\underset{G_2}{\Rightarrow}} p_1$, $w_2 \overset{*}{\underset{G_2}{\Rightarrow}} p_2$, adică $x \overset{*}{\underset{G_2}{\Rightarrow}} p$.

Să observăm că în prima regulă de generare aplicată în deducția considerată nu putem avea $x \in I_N'' \setminus I_N'$. Rămîne încă posibilitatea ca prima regulă de generare să fie de forma $x \to w_1 u$, unde $w_1 \in I_N'$, $u_1 \in I_N'' \setminus I_N'$. Aceasta implică existența în G_2 a regulii $x \to w_1 \ldots w_n$ și în G_3 a regulilor $x \to w_1 u_1$, $u_1 \to w_2 u_2, \ldots, u_{n-3} \Rightarrow w_{n-2}, u_{n-2} \to w_{n-1} w_n$.

Cum $w_1 u_1 \overset{*}{\underset{G_3}{\Rightarrow}} p$, cuvîntul p se poate scrie $p = p_1 p_1'$, unde $w_1 \overset{*}{\underset{G_3}{\Rightarrow}} p_1$, $u_1 \overset{*}{\underset{G_3}{\Rightarrow}} p_1'$. Conform ipotezei inductive $w_1 \overset{*}{\underset{G_2}{\Rightarrow}} p_1$. Apoi, deoarece singura regulă de generare care-l conține în partea stîngă pe u_1 este $u_1 \to w_2 u_2$, p_1' se poate scrie $p_1' = p_2 p_2'$, unde $w_2 \overset{*}{\underset{G_2}{\Rightarrow}} p_2$, $u_2 \overset{*}{\underset{G_3}{\Rightarrow}} p_2'$.

Analog avem $w_2 \overset{*}{\underset{G_2}{\Rightarrow}} p_2$ și $p_2' = p_3 p_3'$ etc.

Astfel $w_j \overset{*}{\underset{G_1}{\Rightarrow}} p_j$, $j = \overline{1, n}$, unde $p = p_1 \dots p_n$ și deci $x \overset{*}{\underset{G_2}{\Rightarrow}} p_1 \dots p_n = p$. ■

Consecința II.5.2. Dacă drumurile unui arbore de deducție pentru gramatica G sub forma normală a lui Chomsky au lungimea maximă egală cu j cuvîntul terminal corespunzător are lungimea cel mult egală cu 2^{j-1}. Dacă pentru $p \in L(G)$ avem $l(p) > 2^{j-1}$ atunci într-un arbore de deducție pentru cuvîntul p există măcar un drum de lungime cel puțin j, cum rezultă imediat ținînd seama de forma specială a regulilor de generare.

Exemplul II.5.5. Fie $G_1 = (\{x_0, x_1, x_2\}, \{i_1, i_2\}, x_0; \{x_0 \to i_2 x_1, x_0 \to i_1 x_2, x_1 \to i_1, x_1 \to i_1 x_0, x_1 \to i_2 x_1 x_1, x_2 \to i_2, x_2 \to i_2 x_0, x_2 \to i_1 x_2 x_2\})$, o gramatică de tip 2. Ne propunem să găsim gramatica G_2 de tip 2 în forma normală a lui Noam Chomsky, echivalentă cu G_1.

Regulile de generare care au deja forma admisă sînt $x_1 \to i_1$, $x_2 \to i_2$. Pentru celelalte reguli de generare vom aplica construcția indicată în demonstrația teoremei, așa cum rezultă din tabelul de mai jos.

Regula de generare din G_1	Regulile de generare corespunzătoare din G_2
$x_0 \to i_2 x_1$	$x_0 \to z_1 x_1$, $z_1 \to i_2$
$x_0 \to i_1 x_2$	$x_0 \to z_2 x_2$, $z_2 \to i_1$
$x_1 \to i_1 x_0$	$x_1 \to z_3 x_0$, $z_3 \to i_1$
$x_1 \to i_2 x_1 x_1$	$x_1 \to z_4 x_1 x_1$, $z_4 \to i_2$
$x_2 \to i_2 x_0$	$x_2 \to z_5 x_0$, $z_5 \to i_2$
$x_2 \to i_1 x_2 x_2$	$x_2 \to z_6 x_2 x_2$, $z_6 \to i_1$

În continuare vom construi gramatica G_3 înlocuind regulile de generare $x_1 \to z_4 x_1 x_1$ și $x_2 \to z_6 x_2 x_2$ respectiv prin $x_1 \to z_4 u_1$, $u_1 \to x_1 x_1$ și $x_2 \to z_6 u_2$, $u_2 \to x_2 x_2$.

Teoremele care urmează constituie rezultate preliminare în vederea prezentării unei alte forme normale pentru gramaticile de tip 2 și anume forma normală datorată lui Sheila Greibach.

Teorema II.5.10. *Fie $G = (I_N, I_T, x_0; F)$ o gramatică de tip 2. Fie $(x \to r_1 x' r_2) \in F$ și $\{x' \to q_1, x' \to q_2, \dots, x' \to q_n\} \subseteq F$ toate regulile de generare care conțin x' în partea stîngă. Atunci $L(G) = L(G_1)$, unde $G_1 = (I_N, I_T, x_0; F_1)$ F_1 fiind obținut din F înlocuind regula de generare $x \to r_1 x' r_2$ prin mulțimea de reguli $x \to r_1 q_1 r_2$, $x \to r_1 q_2 r_2, \dots, x \to r_1 q_n r_2$.*

Demonstrație. Avem $L(G_1) \subseteq L(G)$ deoarece, dacă regula $x \to r_1 q_j r_2$ este folosită într-o deducție din G_1 atunci $x \underset{G}{\Rightarrow} r_1 x' r_2 \underset{G}{\Rightarrow} r_1 q_3 r_2$.

Invers, să arătăm că $L(G) \subseteq L(G_1)$. Dacă în deducția $x \overset{*}{\underset{G}{\Rightarrow}} p$ se utilizează regula $x \to r_1 x' r_2$ atunci în mod necesar se utilizează ulterior și regula $x' \to q_j$ și acești doi pași pot fi înlocuiți printr-unul singur, anume $x \underset{G_1}{\Rightarrow} r_1 q_3 r_2$. ■

Teorema II.5.11. *Fie* $G = (I_N, I_T, x_0; F)$ *o gramatică de tip 2 și* $F' = \{x \to xp_j \mid j = \overline{1, n}\}$, $F'' = \{x \to q_j \mid j = \overline{1, m}\}$ *mulțimile de reguli de generare care conțin* x *în partea stîngă,* F' *fiind constituită din acele reguli în care* x *apare și ca primul simbol din partea dreaptă. Atunci* $L(G) = L(G_1)$, *unde* $G_1 = (I_N \cup \{z\}, I_T, x_0; F_1)$, F_1 *fiind obținut din* F *prin înlocuirea mulțimilor* F' *și* F'' *respectiv prin* F_1' *și* F_1'', *unde*:

$$F_1' = \{z \to p_j,\ z \to p_j z,\ j = \overline{1, n}\},$$

și

$$F_1'' = \{x \to q_j,\ x \to q_j z,\ j = \overline{1, m}\}.$$

Aici $p_j \in I_T^*$, $j = \overline{1, n}$, $q_h \in I_T^*$, $h = \overline{1, m}$.

Demonstrație. În baza teoremei II.5.6 putem utiliza pentru deducerea cuvintelor din limbajele considerate numai deducții imediate la stînga.

Fie $p \in L(G)$. Dintr-o deducție imediată la stînga a cuvîntului p în gramatica G putem construi o deducție a lui p în gramatica G_1 astfel: fiecare apariție în deducția imediată la stînga a șirului

$$r_1 x r_2 \underset{G}{\Rightarrow} r_1 x p_{j_1} r_2 \underset{G}{\Rightarrow} r_1 x p_{j_2} p_{j_1} r_2 \underset{G}{\Rightarrow} \dots \underset{G}{\Rightarrow} r_1 x p_{j_h} \dots p_{j_1} r_2 \underset{G}{\Rightarrow}$$

$$\underset{G}{\Rightarrow} r_1 q_j p_{j_h} \dots p_{j_1} r_2,$$

va fi înlocuită prin:

$$r_1 x r_2 \underset{G_1}{\Rightarrow} r_1 q_j z r_2 \underset{G_1}{\Rightarrow} r_1 q_j p_{j_h} z r_2 \underset{G_1}{\Rightarrow} \dots \underset{G_1}{\Rightarrow} r_1 q_j p_{j_h} \dots p_{j_2} z r_2 \underset{G_1}{\Rightarrow}$$

$$\underset{G_1}{\Rightarrow} r_1 q_j p_{j_h} \dots p_{j_2} p_{j_1} r_2,$$

și astfel $L(G) \subseteq L(G_1)$.

Să observăm că deducția din gramatica G_1 nu este o deducție imediată la stînga.

Să considerăm acum o deducție imediată la stînga a lui p, în G_1. Înlocuim șirul:

$$r_1 x r_2 \underset{G_1}{\Rightarrow} r_1 q_j z r_2 \underset{G_1}{\Rightarrow} r_1 q_j p_{j_h} z r_2 \underset{G_1}{\Rightarrow} \dots \underset{G_1}{\Rightarrow} r_1 q_j p_{j_h} \dots p_{j_2} z r_2 \underset{G_1}{\Rightarrow}$$

$$\underset{G_1}{\Rightarrow} r_1 q_j p_{j_h} \dots p_{j_2} p_{j_1} r_2,$$

din această deducție prin

$$r_1 x r_2 \underset{G}{\Rightarrow} r_1 x p_{j_1} r_2 \underset{G}{\Rightarrow} r_1 x p_{j_2} p_{j_1} r_2 \underset{G}{\Rightarrow} \dots \underset{G}{\Rightarrow} r_1 x p_{j_k} \dots$$

$$\dots p_{j_2} p_{j_1} r_2 \underset{G}{\Rightarrow} r_1 q_j p_{j_k} \dots p_{j_1} r_2 .$$

obținînd astfel o deducție a lui p în $L(G)$. Astfel $L(G_1) \subseteq L(G)$. ■

Teorema II.5.12. *(teorema lui Sheila Greibach de existență a formei normale). Orice limbaj L de tip 2 cu $\varepsilon \notin L$ poate fi generat de o gramatică în care orice regulă de generare este de forma $x \to ir$, unde $x \in I_N$, $i \in I_T$, $r \in I_N^*$.*

Demonstrație. Fie $G = (I_N, I_T, x_0; F)$ o gramatică de tip 2 sub forma normală a lui Chomsky și $I_N = \{x_0, x_1, \dots, x_n\}$. Vom construi efectiv o gramatică care să satisfacă la condițiile cerute și care să genereze limbajul $L(G)$, procedînd în mai multe etape.

Etapa I. Vom modifica regulile de generare astfel încît dacă $(x_j \to x_k r) \in F$, $r \in I_N^*$ să avem $k \geqslant j$, obținînd o gramatică G_1, echivalentă cu G.

Pentru aceasta vom proceda inductiv asupra indicelui j, $j = \overline{1, n}$, pentru $j = 0$ condiția fiind de la sine îndeplinită.

Să presupunem că au fost modificate toate regulile de generare de forma $x_j \to x_k r$ astfel încît $k \geqslant j$ pentru $j = \overline{1, h}$ și gramatica obținută în acest stadiu să fie echivalentă cu G.

Vom modifica acum regulile de generare care au în partea stîngă variabila x_{h+1}. Dacă $x_{h+1} \to x_k r$ este o regulă de generare cu $k < h + 1$ atunci, utilizînd ipoteza inductivă și teorema II.5.10, vom înlocui această regulă cu o mulțime de reguli de forma $x_{h+1} \to x_m r$, proces care se poate repeta pînă cînd $m \geqslant h + 1$ (de cel mult de $h - 1$ ori).

Etapa a II-a. Dacă există reguli de generare de forma $x_j \to x_j q$ vom aplica teorema II.5.11 mulțimii de reguli de generare care au în primul membru pe x_j obținînd o gramatică G_2 echivalentă cu G_1, ale cărei reguli de generare au una din formele:

$$x_j \to x_k r, \quad k > j, \quad r \in I_N^*,$$

$$x_j \to ir, \quad i \in I_T, \quad r \in I_N^*,$$

$$z_j \to q, \quad q \in (I_N \cup \{z_1, \dots, z_n\})^*.$$

Etapa a III-a. Vom modifica regulile de generare astfel ca primul simbol din partea dreaptă să fie un simbol terminal.

Observăm mai întîi că toate regulile de generare care conțin în partea stîngă pe x_n au ca prim simbol în partea dreaptă un terminal. Apoi, primul simbol din partea stîngă a regulilor de generare care încep cu x_{n-1} poate fi sau x_n sau un simbol terminal. În primul caz aplicînd teorema II.5.10 putem înlocui regula respectivă cu noi reguli de generare care au ca prim simbol în partea dreaptă un simbol terminal. Vom proceda similar pentru x_{n-2} etc.

Etapa a IV-a. În sfîrșit, înlocuim regulile de generare care au z în partea stîngă și primul simbol din dreapta neterminal prin reguli în care primul

simbol din partea dreapta este un simbol terminal utilizînd în acest scop teorema II.5.10.■

Construcţia din această teoremă se poate urmări în:

Exemplul II.5.6. Fie gramatica de tip 2, $G = (\{x_0, x_1, x_2\}, \{i_1, i_2\}, x_0;$ $\{x_0 \to x_1 x_2, x_1 \to x_2 x_0, x_1 \to i_2, x_2 \to x_0 x_1, x_2 \to i_1\})$. Urmînd construcţia descrisă în demonstraţia teoremei precedente ne propunem să găsim forma normală de tip Greibach a gramaticii G.

Etapa I. Să observăm că toate regulile de generare care au în partea stîngă x_0 sau x_1 satisfac condiţia cerută. Rămîne să ne ocupăm de regulile care încep cu x_2, anume $x_2 \to x_0 x_1$ care va fi înlocuită pe rînd cu

$$x_2 \to x_1 x_2 x_1,$$

şi apoi prin:

$$x_2 \to x_2 x_0 x_2 x_1 \text{ şi } x_2 \to i_2 x_2 x_1.$$

Obţinem astfel gramatica G_1 echivalentă cu G, $G_1 = (\{x_0, x_1, x_2\}, \{i_1, i_2\},$ $x_0; \{x_0 \to x_1 x_2, x_1 \to x_2 x_0, x_1 \to i_2, x_2 \to x_2 x_0 x_2 x_1, x_2 \to i_2 x_2 x_1, x_2 \to i_1\})$.

Etapa a II-a. Vom aplica teorema II.5.11 regulilor de generare care încep cu x_2 (întrucît apare regula de generare $x_2 \to x_2, \ldots$), introducînd simbolul z şi înlocuind regula de generare $x_2 \to x_2 x_0 x_2 x_1$ prin $z \to x_0 x_2 x_1$ şi $z \to x_0 x_2 x_1 z$, iar regulile $x_2 \to i_2 x_2 x_1$, $x_2 \to i_1$ prin $x_2 \to i_2 x_2 x_1$, $x_2 \to i_2 x_2 x_1 z$, $x_2 \to i_1 z$. Mulţimea de reguli de generare pentru noua gramatică va fi:

$$\begin{array}{lll} x_0 \to x_1 x_2 & x_2 \to i_2 x_2 x_1 & z \to x_0 x_2 x_1 z \\ x_1 \to x_2 x_0 & x_2 \to i_1 & z \to x_0 x_2 x_1 \\ x_1 \to i_2 & x_2 \to i_2 x_2 x_1 z & \\ & x_2 \to i_1 z & \end{array}$$

Etapa a III-a. Toate regulile de generare avînd x_2 în partea stîngă au ca prim simbol un terminal. Apoi, vom folosi aceste reguli de generare în regula $x_1 \to x_2 x_0$ în aşa fel încît primul simbol din partea dreaptă să fie un terminal şi în sfîrşit, folosind ultimele reguli obţinute vom modifica regula $x_0 \to x_1 x_2$. Obţinem astfel următoarea mulţime de reguli de generare.

$$x_2 \to i_2 x_2 x_1;\ x_2 \to i_1;\ x_2 \to i_2 x_2 x_1 z;\ x_2 \to i_1 z;$$

$$x_1 \to i_2;\ x_1 \to i_2 x_2 x_1 x_0;\ x_1 \to i_1 x_0;\ x_1 \to i_2 x_2 x_1 z x_0;$$

$$x_1 \to i_1 z x_0;\ x_0 \to i_2 x_2;\ x_0 \to i_2 x_2 x_1 x_0 x_2;\ x_0 \to i_1 x_0 x_2;$$

$$x_0 \to i_2 x_2 x_1 z x_0 x_2;\ x_0 \to i_1 z x_0 x_2;\ z \to x_0 x_2 x_1 z;\ z \to x_0 x_2 x_1.$$

Etapa a IV-a. Regulile de generare care conţin în partea stîngă pe z au ca prim simbol în partea dreaptă pe x_0. Folosind regulile de generare care conţin pe x_0 în primul membru, cele două reguli $z \to x_0 x_2 x_1 z$ şi $z \to x_0 x_2 x_1$ vor fi înlocuite respectiv prin:

$z \to i_2\, x_2\, x_2\, x_1\, z$ $\qquad$ $z \to i_2\, x_2\, x_2\, x_1$

$z \to i_2\, x_2\, x_1\, x_0\, x_2\, x_2\, x_1\, z$ $\qquad$ $z \to i_2\, x_2\, x_1\, x_0\, x_2\, x_2\, x_1$

$z \to i_1\, x_0\, x_2\, x_2\, x_1\, z$ $\qquad$ $z \to i_1\, x_0\, x_2\, x_2\, x_1$

$z \to i_2\, x_2\, x_1\, zx_0\, x_2\, x_2\, x_1\, z$ $\qquad$ $z \to i_2\, x_2\, x_1\, zx_0\, x_2\, x_2\, x_1$

$z \to i_1\, zx_0\, x_2\, x_2\, x_1\, z$ $\qquad$ $z \to i_1\, zx_0\, x_2\, x_2\, x_1$

Gramatica sub forma normală Greibach, echivalentă cu gramatica dată G este deci $G' = (\{x_0, x_1, x_2, z\}, \{i_1, i_2\}, z_0; F')$, unde F' este alcătuit din următoarele reguli:

$x_2 \to i_2\, x_2\, x_1$ $\qquad$ $x_1 \to i_2$ $\qquad$ $x_0 \to i_2\, x_2$

$x_2 \to i_1$ $\qquad$ $x_1 \to i_2\, x_2\, x_1\, x_0$ $\qquad$ $x_0 \to i_2\, x_2\, x_1\, x_0\, x_2$

$x_2 \to i_2\, x_2\, x_1\, z$ $\qquad$ $x_1 \to i_1\, x_0$ $\qquad$ $x_0 \to i_1\, x_0\, x_2$

$x_2 \to i_1\, z$ $\qquad$ $x_1 \to i_2\, x_2\, x_1\, zx_0$ $\qquad$ $x_0 \to i_2\, x_2\, x_1\, zx_0\, x_2$

$x_1 \to i_1\, zx_0$ $\qquad$ $x_0 \to i_1\, zx_0\, x_2$

$z \to i_2\, x_2\, x_2\, x_1\, z$ $\qquad$ $z \to i_2\, x_2\, x_2\, x_1$

$z \to i_2\, x_2\, x_1\, x_0\, x_2\, x_2\, x_1\, z$ $\qquad$ $z \to i_2\, x_2\, x_1\, x_0\, x_2\, x_2\, x_1$

$z \to i_1\, x_0\, x_2\, x_2\, x_1\, z$ $\qquad$ $z \to i_1\, x_0\, x_2\, x_2\, x_1$

$z \to i_2\, x_2\, x_1\, zx_0\, x_2\, x_2\, x_1\, z$ $\qquad$ $z \to i_2\, x_2\, x_1\, zx_0\, x_2\, x_2\, x_1$

$z \to i_1\, zx_0\, x_2\, x_2\, x_1\, z$ $\qquad$ $z \to i_1\, zx_0\, x_2\, x_2\, x_1$.

Ne vom ocupa în continuare de unele proprietăți de închidere ale limbajelor de tip 2 (v. și §.II.2).

Teorema II.5.13. *Clasa limbajelor de tip 2 este închisă la operația de substituție.*

Demonstrație. Fie $E_1, \ldots, E_r$, E limbaje de tip 2 asupra alfabetului I și $E_j = L(G_j)$ unde $G_j = (I_N^j, \{i_1, \ldots, i_r\}, x_0^j; F^j)$, $j = \overline{1, r}$ și $E = L(G)$ cu $G = (I_N, \{i_1, \ldots, i_r\}, x_0; F)$. Putem presupune, fără a restrînge generalitatea că $I_N^j \cap I_N^k = \varnothing$ pentru $j \neq k$ și $I_N \cap I_N^j = \varnothing$, $j = \overline{1, r}$.

Ne propunem să construim o gramatică de tip 2 care să genereze limbajul $\Phi_{\{E_1, \ldots, E_r\}}(E)$ obținut prin operația de substituție generată de r-uplul de limbaje $\{E_1, \ldots, E_r\}$ (v. definiția II.4.3).

Fie $G_0 = (I_N \cup \bigcup_{j=1}^{r} I_N^j, \{i_1, \ldots, i_r\}, x_0; F)$ cu $F_0 = \bigcup_{j=1}^{r} F^j \cup F'$ unde F' se obține din F prin înlocuirea în fiecare regulă de generare a simbolilor terminali $i_1, \ldots, i_r$ respectiv $x_0^1, \ldots, x_0^r$.

Să arătăm mai întîi că $\Phi_{\{E_1, \ldots, E_r\}}(E) \subseteq L(G_0)$.

Dacă $q \in \Phi_{\{E_1, \ldots, E_r\}}(E)$ există $p \in E$ astfel ca $q = \varphi(p)$ (φ fiind aplicația utilizată în definiția II.4.3). Fie deci $p = i_{j_1} i_{j_2} \ldots i_{j_n}$ unde $j_{j_k} \in \overline{1, r}$. $k = \overline{1, n}$. Urmează că $q = q_{j_1} q_{j_2} \ldots q_{j_n}$ unde $q_{j_k} \in E_{j_k}$, $k = \overline{1, n}$. În gramatica G_0 avem deducția $x_0 \overset{*}{\underset{G_0}{\Rightarrow}} x_0^{j_1} x_0^{j_2} \ldots x_0^{j_n} \overset{*}{\underset{G_0}{\Rightarrow}} q_{j_1} x_0^{j_2} \ldots x_0^{j_n} \overset{*}{\underset{G_0}{\Rightarrow}} \ldots \overset{*}{\underset{G_0}{\Rightarrow}} q$.

Invers, fie $q \in L(G_0)$ adică $x_0 \overset{*}{\underset{G_0}{\Rightarrow}} q$. Există atunci un cuvînt $y \in \{x_0^1, x_0^2, \ldots, x_0^r\}^*$, $y = x_0^{j_1} x_0^{j_2} \ldots x_0^{j_n}$ astfel încît să existe deducțiile $x_0 \overset{*}{\underset{G_0}{\Rightarrow}} y \overset{*}{\underset{G_0}{\Rightarrow}} q$ și $x_0 \overset{*}{\underset{G}{\Rightarrow}} v = i_{j_1} i_{j_2} \ldots i_{j_n}$.

Aceasta revine la a scrie $q = q_{j_1} q_{j_2} \ldots q_{j_n}$ cu $x_0^{j_k} \overset{*}{\underset{G_{j_k}}{\rightarrow}} q_{j_k}$, $k = \overline{1, n}$. Astfel $q = \varphi(p)$ (v. definiția II.4.3) și $q \in \Phi_{\{E_1, \ldots, E_r\}}(E)$. ■

Teorema II.5.14. *Clasa limbajelor de tip 2 este închisă la endomorfismele semigrupului liber I^*.*

Demonstrație. Deoarece orice limbaj finit este limbaj regulat, deci și de tip 2, afirmația se obține ținînd seama de faptul că dacă $E_j = \{i_{\alpha(j)}\}$ unde $\alpha : \overline{1, r} \to \overline{1, r}$, atunci aplicația φ din definiția II.4.3 este un endomorfism pe I^* și $\Phi_M(E) = \varphi(E)$ și se utilizează și teorema precedentă. ■

Teorema II.5.15. *Dacă E este un limbaj de tip 2 și h este un endomorfism al semigrupului liber I^* care păstrează lungimea cuvintelor atunci $h^{-1}(E)$ este un limbaj de tip 2.*

Demonstrație. Aplicația $h^{-1} : I^* - \mathcal{P}(I^*)$ face să corespundă unui simbol $i \in I$ mulțimea finită $h^{-1}(i) \subseteq I$. Atunci $h^{-1}(i)$ este un limbaj regulat și $h^{-1}(E) = \Psi(h^{-1}(i_1), \ldots, h^{-1}(i_r), E)$. ■

Teorema II.5.16. *Clasa limbajelor de tip 2 nu-i închisă la intersecție și complementariere.*

Demonstrație. Vom arăta că există $E_1, E_2 \in \mathcal{L}_2$ și $E_1 \cap E_2 \notin \mathcal{L}_2$. Fie gramaticile :

$$G_1 = (\{x_0, x_1\}, \{i_1, i_2, i_3\}, x_0 ; \{x_0 \to x_0 i_3, x_0 \to x_1 i_3, x_1 \to i_1 i_2, x_1 \to i_1 x_1 i_2\}),$$

și

$$G_2 = (\{x_0, x_1\}, \{i_1, i_2, i_3\}, x_0 ; \{x_0 \to i_1 x_0, x_0 \to i_1 x_1, x_1 \to i_2 i_3, x_1 \to i_2 x_1 i_3\}).$$

Avem $L(G_1) = \{i_1^m i_2^n i_3^n \mid m, n \geqslant 1\}$ și $L(G_2) = \{i_1^m i_2^n i_3^n \mid m, n \geqslant 1\}$. Evident $E = L(G_1) \cap L(G_2) = \{i_1^n i_2^n i_3^n \mid n \geqslant 1\}$. Să presupunem că $E \in \mathcal{L}_2$. Există atunci o gramatică de tip 2, $G = (I_N, \{i_1, i_2, i_3\}, x_0 ; F)$ și $E = L(G)$.

Fie $x \neq x_0$, $x \in I_N$ și $x \overset{*}{\underset{G}{\Rightarrow}} p \in \{i_1, i_2, i_3\}^*$. Dacă această deducție este unică considerăm gramatica $G' = (I_N \setminus \{x\}, \{i_1, i_2, i_3\}, x_0 : F')$ unde F' se obține din F îndepărtînd toate regulile de generare de forma $x \to u$ și înlocuind fiecare apariție a lui x în partea dreaptă a unei reguli de generare din F prin cuvîntul p. Evident $L(G) = L(G')$.

Deci putem presupune că pentru orice simbol neterminal (în particular și pentru x_0) există cel puțin două deducții de forma $x \overset{*}{\underset{G}{\Rightarrow}} p \in \{i_1, i_2, i_3\}^*$. Fie k maximul lungimii cuvintelor care apar în partea dreaptă a unei astfel de deducții. Pe de altă parte, cum $i_1^{k+1} i_2^{k+1} i_3^{k+1} \in E = L(G)$ există deducția $x_0 \overset{*}{\underset{G}{\Rightarrow}} p_1 x p_2 \Rightarrow p_1 q p_2 = i_1^{k+1} i_2^{k+1} i_3^{k+1}$. Astfel $(x \to q) \in F$, $l(q) \leqslant k$ și mai există încă cel puțin un cuvînt terminal $q' \neq q$ astfel că $x \overset{*}{\underset{G}{\Rightarrow}} q'$. Prin urmare există un subcuvînt q al cuvîntului $i_1^{k+1} i_2^{k+1} i_3^{k+1}$ cu $l(q) \leqslant k$ și $q' \neq q$ astfel încît cînd înlocuim pe q prin q' în cuvîntul $i_1^{k+1} i_2^{k+1} i_3^{k+1}$ cuvîntul rezultat aparține limbajului E, ceea ce este evident imposibil, deoarece acest cuvînt nu are forma prescrisă. Prin urmare $E \notin \mathcal{L}_2$.

În consecință clasa $\mathcal{L}_2$ nu este închisă nici la complementariere deoarece în caz contrar fiind închisă la reuniune ar rezulta și închiderea la intersecție. ■

Teorema II.5.17. *Fie $E \in \mathcal{L}_2$. Există $n_1, n_2 \in N$ astfel încît, dacă $p \in E$ și $l(p) > n_1$ atunci $p = tuqvw$, cu $l(uqv) \leqslant n_2$, u și v nu pot fi ambii egali cu e astfel încît pentru orice $h \geqslant 0$ să avem: $tu^h qv^h w \in E$.*

Demonstrație. Fie $G = (I_N, I_T, x_0; F)$ o gramatică de tip 2 în forma normală a lui Chomsky și $E = L(G)$. Dacă $|I_N| = k$ atunci fie $n_1 = 2^{k-1}$ și $n_2 = 2^k$.

Fie $p \in E$ și $l(p) > 2^{k-1}$. Conform consecinței II.5.2 în orice arbore de deducție pentru cuvîntul p există un drum de lungime cel puțin k. Să considerăm într-un asemenea arbore un drum de *lungime maximă*. Acest drum va conține două noduri s_1, s_2 astfel încît:

1) Nodurile au aceeași etichetă x.

2) Nodul s_1 este mai apropiat de rădăcină decît s_2.

3) Drumul de la s_1 la nodul final (conținut obligatoriu de acest drum de lungime maximă) este cel mult de lungime $k+1$.

Într-adevăr, din ultimele $k+2$ noduri ale drumului de lungime maximă numai nodul terminal este etichetat cu un simbol din I_T. Celelalte $k+1$ noduri sînt etichetate cu simboli din I_N și cum $|I_N| = k$ în mod necesar două noduri au aceeași etichetă și aceste două noduri satisfac condițiilor 1)–3).

Atunci, subarborele cu rădăcina în nodul s_1 reprezintă o deducție a unui subcuvînt p_1 de lungime cel mult $2^k = n_2$.

Dacă subcuvîntul p_2 a rezultat în urma deducției descrisă de subarborele cu rădăcina în s_2 atunci $p_1 = r_1 p_2 r_2$ unde r_1 și r_2 nu pot fi concomitent egali

cu cuvîntul nul deoarece prima regulă de generare folosită în subarborele cu rădăcina în s_1 este de forma $x \to x_1 x_2$. x_1 , $x_2 \in I_N$.

Avem

$$x \underset{G}{\overset{*}{\Rightarrow}} r_1 x r_2 \underset{G}{\overset{*}{\Rightarrow}} r_1 p_2 r_2 = p_1 ,$$

unde $l(r_1 p_2 r_2) \leqslant 2^k = n_2$. De aici rezultă că pentru orice $h \geqslant 0$ avem :

$$x \underset{G}{\overset{*}{\Rightarrow}} r_1^h x r_2^h \underset{G}{\overset{*}{\Rightarrow}} r_1^h p_2 r_2^h .$$

Cum cuvîntul p se poate scrie $p = lp_1 w = lr_1 p_2 r_2 w$ rezultă că afirmația făcută este adevărată punînd $u = r_1$, $v = r_2$, $q = p_1$. ■

Observația II.5.4. Teorema precedentă permite ca pentru limbajele de tip 2, infinite pornind de la un anumit cuvînt p din limbaj să construim cuvinte din limbaj, oricît de lungi. Din acest motiv această teoremă este cunoscută sub denumirea de „lema de pompare" (v. Arbib [1]).

Teorema II.5.18. *Problema finitudinii limbajului general de o gramatică G de tip 2 este decidabilă.*

Demonstrație. Fie n_1 , n_2 numerele definite în teorema precedentă și $p \in L(G)$ cu $l(p) > n_1$. Atunci $p = tuqvw$ și $tu^h qv^h w \in L(G)$, $\forall h \geqslant 0$. Deci $l(u) + l(v) > 0$.

Prin urmare, dacă există $p \in L(G)$ cu $l(p) > n_1$ atunci $L(G)$ este infinită.

Să presupunem acum că $L(G)$ este infinită. Atunci există un cuvînt oricît de lung în $L(G)$, în particular există $p \in L(G)$ cu $l(p) > n_1 + n_2$, $p = tuqvw$ cu $l(uqu) \leqslant n_2$, $l(u) + l(v) > 0$ și $tu^h qv^h w \in L(G)$, $\forall h \geqslant 0$.

În particular, pentru $h = 0$ avem $tqw \in L(G)$ și $l(tqw) < l(tuqvw)$ și $l(tqw) > n_1$. Dacă $l(tqw) > n_1 + n_2$ repetăm procedeul pînă cînd găsim un cuvînt din $L(G)$ de lungime l cu $n_1 < l \leqslant n_1 + n_2$. Astfel $L(G)$ este infinit dacă și numai dacă există $p \in L(G)$ cu $n_1 < l(p) \leqslant n_1 + n_2$.

Cum numărul acestor cuvinte este finit putem decide într-un număr finit de pași dacă $L(G)$ este finit sau nu. ■

§ II. 6. CARACTERIZAREA LIMBAJELOR DE TIP 3 ÎN CLASA LIMBAJELOR DE TIP 2

Se știe (v. observația II.2.2) că orice limbaj de tip 3 este și de tip 2, iar existența unor limbaje de tip 2 care nu sînt limbaje regulate va rezulta din acest paragraf, în care ne propunem să facem individualizarea limbajelor de tip 3 în clasa limbajelor de tip 2.

Definiția II.6.1. *O gramatică de tip 2,* $G = (I_N, I_T, x_0; F)$ *are proprietatea de autoimersiune* dacă există $x \in I_N$ astfel încît $x \in pxq$, unde $p, q \in (I_N \cup I_T)^* \setminus \{e\}$.

Un limbaj de tip 2 are proprietatea de autoimersiune dacă el este generat de o gramatică de tip 2 cu proprietatea de autoimersiune.

Observația II.6.1. O gramatică de tip 3 nu are proprietatea de autoimersiune. Într-adevăr, ținînd seama de definiția II.2.1, deducțiile într-o gramatică de tip 3 au forma $x \overset{\bullet}{\Rightarrow} px_1$, unde $x, x_1 \in I_N$, $p \in I_T^*$, sau $x \overset{\bullet}{\Rightarrow} p$ cu $x \in I_N$, $p \in I_T^*$.

Vom arăta printr-un exemplu că există gramatici de tip 2 cu proprietatea de autoimersiune.

Exemplul II.6.1. Fie gramatica de tip 2, $G = (\{x_0, x_1\}, \{i_1, i_2\}, x_0; F)$, unde $F = \{x_0 \to i_1 x_0 x_1 i_2,\ x_0 \to i_1,\ x_1 \to i_2\}$. Se observă că G este cu autoimersiune, deoarece există regula $x_0 \to i_1 x_0 x_1 i_2$.

Teorema II.6.1. *Un limbaj de tip 2 este și de tip* 3, *dacă și numai dacă nu are proprietatea de autoimersiune.*

Demonstrație. Conform observației II.6.1 gramaticile de tip 3 nu au proprietatea de autoimersiune.

Invers, să presupunem că gramatica $G = (I_N, I_T, x_0; F)$ este de tip 2 și nu are proprietatea de autoimersiune.

Putem presupune că oricare ar fi $x \in I_N$ există $p, q \in (I_N \cup I_T)^*$ astfel încît să existe deducția $x_0 \overset{\bullet}{\underset{G}{\Rightarrow}} pxq$ deoarece în caz contrar renunțarea la x și la deducțiile care încep cu x nu modifică $L(G)$.

Să presupunem mai întîi că oricare ar fi $x \in I_N$ există $r_1, r_2 \in (I_N \cup I_T)^*$ astfel ca să avem o deducție $x \overset{\bullet}{\underset{G}{\Rightarrow}} r_1 x_0 r_2$. Să observăm că în acest caz regulile de generare care conțin un simbol neterminal în partea dreaptă pot avea una din formele:

(II.6.1) $\qquad x \to pyq$

(II.6.2) $\qquad x \to py$

(II.6.3) $\qquad x \to qy$

(II.6.4) $\qquad x \to y$

unde $x, y \in I_N$, $p, q \in (I_N \cup I_T)^* \setminus \{e\}$.

Dacă F conține o regulă de generare de forma (II.6.1) atunci se obține deducția:

$$x \Rightarrow pyq \overset{\bullet}{\underset{G}{\Rightarrow}} pp_1 x_0 q_1 q \overset{\bullet}{\underset{G}{\Rightarrow}} pp_1 p_2 xq_2 q_1 q,$$

deci G are proprietatea de autoimersiune, ceea ce contrazice ipoteza.

Dacă F conține simultan reguli de tipul (II.6.2) și (II.6.3) atunci obținem o deducție de forma :

$$x \Rightarrow py \underset{G}{\overset{*}{\Rightarrow}} pp_1 x_0 q_1 \underset{G}{\overset{*}{\Rightarrow}} pp_1 p_2 x' q_2 q_1 \underset{G}{\overset{*}{\Rightarrow}} pp_1 p_2 y' q q_2 q_1$$

$$\underset{G}{\overset{*}{\Rightarrow}} pp_1 p_2 p_3 x_0 q_3 q q_2 q_1 \underset{G}{\overset{*}{\Rightarrow}} pp_1 p_2 p_3 p_4 x q_4 q_3 q q_2 q_1 ,$$

ceea ce de asemenea implică existența proprietății de autoimersiune.

Dacă F ar conține o regulă de generare de forma (II.6.2) atunci cuvîntul $p \in I_T^*$ deoarece în caz contrar F ar conține reguli de tipul (II.6.1) sau (II.6.3), ceea ce antrenează proprietatea de autoimersiune.

Conchidem că în cazul considerat F conține reguli de generare de forma $x \to py$ cu $p \in I_T^*$ sau $x \to y$, deci G este de tip 3.

Dacă F ar conține o regulă de generare de tipul (II.6.3) (fără a conține și reguli de generare de forma (II.6.1) sau (II.6.2) deoarece atunci G ar avea proprietatea de autoimersiune) atunci $q \in I_T^*$. Într-adevăr, dacă q ar conține și simboli neterminali această regulă ar apare ca o regulă de tip (II.6.1) sau (II.6.2).

Fie deci $(x \to yq) \in F$ cu $q \in I_T^*$. Atunci limbajul $\widetilde{L(G)}$ (obținut prin operația de oglindire) este de tip 3 și deoarece clasa limbajelor de tip 3 este închisă la operația de oglindire și limbajul $L(G) = \widetilde{\widetilde{L(G)}}$ este de tip 3.

Limbajul $L(G)$ este de tip 3 și în cazul în care F nu conține reguli de generare de forma (II.6.1 – 3).

Să presupunem acum că există $x \in I_N$ astfel încît să nu apară nici o deducție de forma $x \underset{G}{\overset{*}{\Rightarrow}} r_1 x_0 r_2$. Vom demonstra prin inducție după $n = |I_N|$ că $L(G)$ este de tip 3.

Pentru $n = 1$, $L(G)$ este de tip 3 deoarece neexistînd deducții de forma $x_0 \underset{}{\overset{*}{\Rightarrow}} r_1 x_0 r_2$, F conține un număr finit de reguli de generare de forma $x_0 \to p$, $p \in I_T^*$ și deci $|L(G)| < \infty$.

Să presupunem afirmația adevărată pentru $n = h$ și fie $|I_N| = h + 1$. Considerăm gramaticile $G_1 = (I_N \setminus \{x_0\}, I_T, x; F_1)$ unde F_1 se obține din F prin eliminarea regulilor de generare care conțin pe x_0 și $G_2 = (I_N \setminus \{x\}, I_T \cup \{x\}, x_0; F_2)$, unde F_2 se obține din F prin înlăturarea tuturor regulilor, care conțin pe x în partea stîngă (adică G_2 se obține din G considerîndu-l pe x ca un simbol terminal).

Limbajul $L(G_1)$ este de tip 3 prin ipoteza inductivă, iar $L(G_2)$ este de același tip așa cum rezultă din prima parte a demonstrației. Avem $L(G) = L \cup (L(G_2) \cap I_T^*)$, unde L este rezultatul substituției lui x în $L(G_2)$ prin $L(G_1)$.

Într-adevăr, dacă $p \in L(G_2) \cap I_T^*$ avem evident $p \in L(G)$, iar dacă $p \in L$ atunci $p = r_1 p_1 r_2 p_2 \dots r_k$, unde $x_0 \overset{*}{\underset{G_2}{\Rightarrow}} r_1 x r_2 x \dots x r_k$ și $x \overset{*}{\underset{G_1}{\Rightarrow}} r_1 x_1 r_2 x \dots$ $\dots x r_k$, și $x \overset{*}{\underset{G_1}{\Rightarrow}} p_1, \dots, x \overset{*}{\underset{G_1}{\Rightarrow}} p_{k-1}$. Deci avem în G deducția $x_0 \overset{*}{\Rightarrow} r_1 x r_2 x \dots x r_k \overset{*}{\Rightarrow}$ $\overset{*}{\Rightarrow} r_1 p_1 r_2 x \dots x r_k \overset{*}{\Rightarrow} r_1 p_1 r_2 p_2 \dots p_{k-1} r_k = p$, de unde $L \cup (L(G) \cap I_T^*) \subseteq$ $\subseteq L(G)$.

Dacă $p \in L(G)$ și în deducția $x_0 \overset{*}{\underset{G}{\Rightarrow}} p$ nu apare simbolul x atunci $x_0 \overset{*}{\underset{G_1}{\Rightarrow}} p$ și $p \in L(G_2) \cap I_T^*$. Dacă în deducția $x_0 \overset{*}{\underset{G}{\Rightarrow}} p$ apare la un moment dat x această deducție se poate scrie

$$x_0 \overset{*}{\Rightarrow} q_1 x q_2 x \dots x q_t \Rightarrow p,$$

unde q_j, $j = \overline{1, t}$ nu conțin pe x și regulile de generare utilizate în deducția $x_0 \overset{*}{\underset{G}{\Rightarrow}} q_1 x q_2 x \dots x q_t$ nu au x în partea stîngă. Prin urmare $x_0 \overset{*}{\underset{G_2}{\Rightarrow}} q_1 x q_2 x \dots x q_t$.

În continuare, dat fiind că nu există deducții de forma $x \overset{*}{\underset{G}{\Rightarrow}} r_1 x_0 r_2$ trebuie aplicate în mod necesar reguli de generare care fac parte din F_1 și deci avem $p = q_1 p_1 q_2 p_2 \dots p_{t-1} q_t$, unde $x \overset{*}{\underset{G_1}{\Rightarrow}} p_1, \dots, x \overset{*}{\underset{G_1}{\Rightarrow}} p_{t-1}$.

Deci $p \in L$ și $L(G) \subseteq L \cup (L(G_2) \cap I_T^*)$.

Limbajul $L \cup (L(G_2) \cap I_T^*)$ este un limbaj de tip 3 în baza proprietăților de care se bucură clasa de limbaje de tip 3. ■

Observația II.6.2. Teorema precedentă dă o caracterizare a limbajelor regulate în clasa limbajelor de tip 2.

Observația II.6.3. Există incluziunea strictă $\mathcal{L}_2 \supset \mathcal{L}_3$ cum rezultă din teorema precedentă și exemplul II.6.1.

Vom încheia acest paragraf punînd în evidență o proprietate de închidere relativă la familiile $\mathcal{L}_2$ și $\mathcal{L}_3$.

Am arătat (v. teorema II.5.16) că familia de limbaje $\mathcal{L}_2$ nu este închisă la intersecție. Are loc, însă :

Teorema II.6.2. *Intersecția unui limbaj de tip* 2 *cu un limbaj de tip* 3 *este un limbaj de tip* 2.

Demonstrație. Fie $E_1 \in \mathcal{L}_2$ și $E_2 \in \mathcal{L}_3$ limbaje peste același alfabet I_T. Fie $G_1 = (I_N, I_T, x_0; F)$ gramatica de tip 2 cu $E_1 = L(G_1)$ și $A_2 = (I_T, S, f; s_0)$ semiautomatul în care limbajul E_2 este reprezentat prin mulțimea de stări $S_2 \subseteq S$. Să presupunem mai întîi că $e \notin E_1$ și $e \notin E_2$. Conform teoremei II.5.1 putem presupune că $(x \to e) \notin F$, $\forall x \in I_N$. Fie $S_2 = \{s^1, \dots, s^h\}$

și E_2^j limbajul reprezentat în semiautomatul A_2 prin mulțimea $\{s^j\}$, $j = \overline{1, h}$. Este clar că $E_1 \cap E_2 = E_1 \cap (\bigcup_{j=1}^{h} E_2^j) = \bigcup_{j=1}^{h} (E_1 \cap E_2^j)$.

Limbajul $E_1 \cap E_2^j$ este generat de gramatica de tip 2, $G^j = ((I_N \cup I_T) \times \times S \times S, I_T, (x_0, s_0, s^j); F')$ unde F' constă din toate regulile de generare având una din următoarele forme:

a) $(x, s, s') \to (y_1, s, s_1)(y_2, s_1, s_2) \dots (y_n, s_{n-1}, s')$, unde $s, s', s_j \in S$, $j = \overline{1, n-1}$ și $(x \to y_1 y_2 \dots y_n) \in F$, $y_j \in I_N \cup I_T$,

b) $(x, s, s') \to x$, unde $x \in I_T$ și $f(s, x) = s'$.

Fie $p \in E_1 \cap E_2^j$. Atunci $x_0 \overset{*}{\Rightarrow} p$ și $f(s_0, p) = s^j$. Fie $p = i_1 i_2 \dots i_n$, $s_h = f(s_0, i_1 i_2 \dots i_h)$, $h = \overline{1, n-1}$. Vom arăta că $p \in L(G^j)$. Într-adevăr, putem considera deducția $(x_0, s_0, s^j) \overset{*}{\underset{G^j}{\Rightarrow}} (i_1, s_0, s_1)(i_2, s_1, s_2) \dots (i_n, s_{n-1}, s^j)$ din gramatica G^j. Avem de asemenea regulile de generare $(i_1, s_0, s_1) \to i_1, \dots, (i_n, s_{n-1}, s^j) \to i_n$. Aplicîndu-le deducția de mai sus obținem deducția $(x_0, s_0, s^j) \overset{*}{\Rightarrow} p$, adică $p \in L(G^j)$.

Invers, fie $p \in L(G^j)$. Prima regulă de generare utilizată este $(x_0, s_0, s^j) \to (y_1, s_0, s_1')(y_2, s_1', s_2') \dots (y_k, s_{k-1}', s_k')$. În continuare vom aplica numai regulile de generare indicate în tipul a). Vom obține în mod necesar deducția $(x_0, s_0, s^j) \overset{*}{\underset{G^j}{\Rightarrow}} (i_1, s_0, s_1)(i_2, s_1, s_2) \dots (i_n, s_{n-1}, s^j)$. Pentru a obține cuvîntul p este obligatorie utilizarea regulilor de forma b) și anume: $(i_1, s_0, s_1) \to i_1, \dots, (i_n, s_{n-1}, s^j) \to i_n$. Existența acestor reguli de generare implică $f(s_0, i_1) = s_1$, $f(s_1, i_2) = s_2, \dots, f(s_{n-1}, i_n) = s^j$, deci $p = i_1 i_2 \dots i_n \in E_1 \cap E_2^j$.

Astfel $L(G^j) = E_1 \cap E_2^j$ și deci $E_1 \cap E_2^j$ este un limbaj de tip 2. Cum clasa limbajelor de tip 2 este închisă la reuniune rezultă că $E_1 \cap E_2$ este un limbaj de tip 2.

Dacă numai unul din limbaje conține cuvîntul nul atunci construim o gramatică G' de tip 2 cu simbolul inițial x_0 care generează limbajul $(E_1 \setminus \{e\}) \cap (E_2 \setminus \{e\})$ așa cum am procedat în prima parte a demonstrației și $E_1 \cap E_2 = L(G')$.

Dacă $e \in E_1 \cap E_2$ (adică ambele limbaje conțin cuvîntul nul) atunci adăugăm gramaticii G' un nou simbol inițial x_0' și regulile de generare $x_0' \to e$ și $x_0' \to x_0$. Noua gramatică generează limbajul $E_1 \cap E_2$. ■

§ II. 7. LIMBAJE DE TIP ALGOL

Limbajele utilizate în „conversația" cu mașinile electronice de calcul (așa-numitele limbaje de programare), sînt intermediare între limbajul științific utilizat în mod curent de om și „limbajul-mașină" utilizat în interiorul

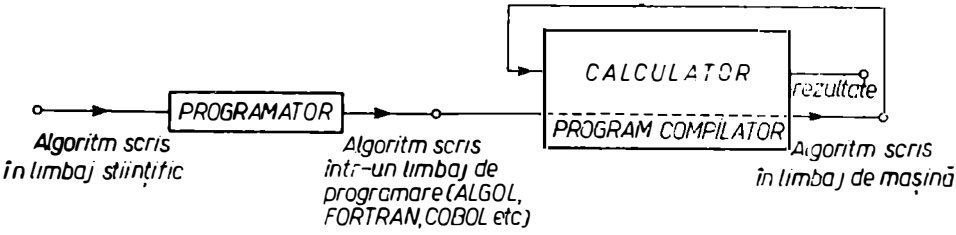

Fig. II. 7.1

calculatorului, motiv pentru care împrumută caracteristici ale ambelor tipuri de limbaje.

Sarcina traducerii unui text scrie în limbajul științific (care descrie algoritmul rezolvării unei probleme) în limbajul de programare revine programatorului.

Sarcina traducerii textului scris în limbajul de programare în limbajul-mașină revine calculatorului însuși care o execută cu ajutorul unui program special numit compilator[1]. Textul astfel prelucrat este iar preluat de calculator și executat, mașina oferind în final rezultatele problemei.

Schema acestei activități este reprezentată în figura II.7.1.

Deoarece traducerea din limbajul de programare în limbajul-mașină se efectuează de către calculator rezultă că limbajul de programare trebuie să posede o descriere precisă, conținută într-un număr finit de reguli.

Aceste reguli se pot da în parte în forma *definițiilor de tip Backus* numite și *definiții recursive* care sînt de fapt ecuații pe mulțimea unor limbaje (înțelegînd aici prin limbaj o parte a unui semigrup liber). Astfel, limbajele care sînt soluții ale unor sisteme de ecuații abstractizează proprietățile limbajelor de programare.

Pentru descrierea unui limbaj de programare (adică pentru formularea definițiilor despre care a fost vorba) se folosește un limbaj mai larg numit *metalimbaj*. Simbolurile care fac parte din acesta le vom numi *simboluri metalingvistice*. Acestea sînt :

1) litere mari și mici ale alfabetului latin
2) cifrele arabe
3) parantezele obișnuite „(“ și „)“
4) parantezele unghiulare „<“ și „>“ care servesc la delimitarea cuvintelor care descriu noțiuni ale limbajului de programare
5) simbolul „: : =“, care are semnificația, „este definit ca“
6) simbolul „ | “ care are semnificația „sau“.

Iată un exemplu de definiție în metalimbaj :

---

[1] Compilatorul este un program foarte lung care necesită o memorie de dimensiuni mari, specifică calculatoarelor din generația a III-a (calculatoare cu memorie cu ferită și tranzistori).

$<$literă$>$: : = $A \mid B \mid C \mid D \mid E \mid F \mid G \mid H \mid I \mid J \mid K \mid L \mid M \mid N \mid O \mid P \mid Q \mid R \mid S \mid T \mid U \mid V \mid W \mid X \mid Y \mid Z \mid$
$<$ cifră $>$: : = $0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

Vom da acum definiția a două noțiuni comune unor limbaje de programare (ALGOL, FORTRAN etc.)

$<$ identificator $>$: : = $<$ literă $> \mid <$ identificator $> <$ literă $> \mid$
$<$ identificator $> <$ cifră $>$
$<$ întreg fără semn $>$: : = $<$ cifră $> \mid <$ întreg fără semn $> <$ cifră $>$

Fie alfabetul $I = \{A, B, C,..., Y, Z, 0, 1,..., 9\}$. Să considerăm limbajele $E_1 = \{A,....Z\}$ și $E_2 = \{0,...,9\}$. Fie Z și Y limbajele alcătuite respectiv din identificatori și întregi fără semn. Definițiilor de mai sus le corespund ecuațiile :

$$Z = E_1 \cup ZE_1 ,$$

$$Y = E_2 \cup YE_2 .$$

Trecem acum la prezentarea formală a noțiunilor descrise mai sus.

Fie $I = \{y_1, ..., y_m\}$ un alfabet finit și $E_1, ..., E_n \subseteq I^*$ o mulțime finită de limbaje. Să considerăm alfabetul $K = \{y_1, ..., y_m, E_1, ..., E_n\}$.

Definiția II.7.1. Un *term* este un cuvînt din K^*. Simbolii dintr-un term sînt fie cuvinte din semigrupul I^* *coeficienți ai termului*, fie simboli de forma E_j, $j = \overline{1, n}$ *variabile ale termului.*

Limbajul descris de un term $T = p_1 E_{j_1} p_2 E_{j_2} ... E_{j_t} p_{t+1}$ este limbajul $\{p_1 r_{j_1} p_2 r_{j_2} ... r_{j_t} p_{t+1} \mid r_{j_k} \in E_{j_k}, k = \overline{1, t}\}$, pe care-l vom nota cu $T(E_1, ..., E_n)$.

Definiția II.7.2. O funcție $h : \mathcal{P}_f(I^*)^n \to \mathcal{P}_f(I^*)$ [1] se numește *funcție standard peste alfabetul* I, dacă :

$$h(E_1, ..., E_n) = \bigcup_{j=1}^{n} T_j(E_1, ..., E_n)$$

T_j fiind termi din K^*, avînd ca variabile limbaje finite.

Ne vom ocupa în continuare de sisteme de ecuații de forma :

$$\text{(II.7.1)} \qquad \begin{cases} E_1 = H_1(E_1, ..., E_n) \\ E_n = H_n(E_1, ..., E_n) \end{cases}$$

unde $H_j(E_1, ..., E_n)$ sînt funcții standard peste alfabetul I.

Folosind notația vectorială sistemul (II.7.1) se poate scrie sub forma :

$$\text{(II.7.2)} \qquad \mathbf{E} = \mathbf{H}(\mathbf{E}),$$

unde $\mathbf{E} = (E_1, ..., E_n)$, $\mathbf{H}(\mathbf{E}) = (H_1(E_1, ..., E_n), ..., H_n(E_1, ..., E_n))$

1) cu $\mathcal{P}_f(I^*)$ am notat mulțimea limbajelor finite din $\mathcal{P}(I^*)$.

Se spune că **H** este o funcție vectorială standard dacă toate componentele sale sînt funcții standard.

Teorema II.7.1. *Dacă* $\mathbf{E} \subseteq \mathbf{E}'$ *și* **H** *este o funcție vectorială standard atunci* $\mathbf{H}(\mathbf{E}) \subseteq \mathbf{H}(\mathbf{E}')$, *adică funcțiile vectoriale standard sînt monotone.*

Demonstrație. Pentru simplitate vom demonstra afirmația în cazul cînd **H** are o singură componentă, deoarece demonstrarea în cazul general revine la același raționament aplicat fiecărei componente în parte.
Astfel

$$\mathbf{H}(\mathbf{E}) = \mathbf{H}(E_1, ..., E_n) = \bigcup_{k=1}^{h} T_k(E_1, ..., E_n) \subseteq \bigcup_{k=1}^{h} T_k(E_1', ..., E_n') =$$

$$= \mathbf{H}(E_1', ..., E_n') = \mathbf{H}(\mathbf{E}')$$

aceasta, deoarece ținînd seama de forma unui term avem evident $T_k(E_1, ..., E_n) \subseteq T_k(E_1', ..., E_n')$ ■

Teorema II.7.2. *Fie* $\mathbf{D}^{(h+1)} = (D_1^{(h+1)}, ..., D_n^{(h+1)}) = \mathbf{H}(D^{(h)})$, $\mathbf{D}^{(0)} = (D_1^{(0)}, ..., D_n^{(0)}) = \mathbf{H}(\varnothing, ..., \varnothing)$, *unde* **H** *este o funcție vectorială standard. Atunci*

$$\mathbf{D} = \bigcup_{h=0}^{\infty} \mathbf{D}^{(h)}, \tag{II.7.3}$$

este o soluție a sistemului (II.7.2)[1]

Demonstrație. Vom verifica, prin inducție că $\mathbf{D}^{(h-1)} \subseteq \mathbf{D}^{(h)}$, $\forall h \in N$, unde $\mathbf{D}^{(-1)} = (\varnothing, ..., \varnothing)$. Avem, evident pentru $h = 0$, $(\varnothing, ..., \varnothing) \subseteq \mathbf{D}^0$. Apoi, să presupunem că $\mathbf{D}^{(k-1)} \subseteq \mathbf{D}^{(k)}$ pentru $k \leqslant h$. Rezultă:

$$\mathbf{D}^{(h+1)} = \mathbf{H}(\mathbf{D}^{(h)}) \supseteq \mathbf{H}(\mathbf{D}^{(h-1)}) = \mathbf{D}^{(h)},$$

conform teoremei II.7.1.

Mai mult, observăm că pentru $k_1 \leqslant k_2$ avem $\mathbf{D}^{(k_1)} \subseteq \mathbf{D}^{(k_2)}$.

Fie deci

$$D_j = \bigcup_{h=0}^{\infty} D_j^{(h)},$$

și să calculăm a j-a componentă din $\mathbf{H}(\mathbf{D})$:

$$H_j(\mathbf{D}) = H_j\left(\bigcup_{h=0}^{\infty} \mathbf{D}^{(h)}\right) = H_j\left(\bigcup_{h_1=0}^{\infty} D_1^{(h_1)}, ..., \bigcup_{h_n=0}^{\infty} D_n^{(h_n)}\right) =$$

$$= \bigcup_{(h_1, h_2, ..., h_n) \in N^n} H_j(D_1^{(h_1)}, ..., D^{(h_n)}) = \bigcup_{h=0}^{\infty} H_j(D_1^{(h)}, ..., D_n^{(h)}) =$$

$$= \bigcup_{h=0}^{\infty} D_j^{(h+1)} = \bigcup_{h=1}^{\infty} D_j^{(h)} = \bigcup_{h=0}^{\infty} D_j^{(h)} = D_j.$$

---

[1] În mulțimea soluțiilor sistemului (II.7.2) (care este nevidă deoarece orice aplicație izotonă $\varkappa : M \to M$, unde M este o latice completă are un punct fix — teorema lui Knäster-Taraski soluția (II.7.3) este primul element (v. teorema II.7.6).

Am folosit aici faptul că $H_j(D_1^{(h_1)}, ..., D_n^{(h_n)}) \subseteq H_j(D_1^{(h)}, ..., D_n^{(h)})$, unde $h = \max\{h_1, ..., h_n\}$. ■

Definiția II.7.3. *Limbajele de tip* ALGOL[1] sînt componentele soluției (II.7.3) a sistemului (II.7.2).

Teorema II.7.3. *Limbajele de tip* ALGOL *sînt limbaje de tip* 2.

Demonstrație. Fie $\mathbf{E} = \mathbf{H}(\mathbf{E}) = (H_1(E_1, ..., E_n), ..., H_n(E_1, ..., E_n))$ un sistem în care $\mathbf{H}$ este o funcție vectorială standard și $H_j(E_1, ..., E_n) = \bigcup_{k=1}^{n(j)} T_{jk}(E_1, ..., E_n)$ exprimarea componentelor lui $\mathbf{H}$ ca reuniune de termi. Astfel $T_{jk} \in K^*$, unde $K = I \cup \{E_1, ..., E_n\}$[2]. Avem $T_{jk}(E) = t_{jk1}(E) ... t_{jkm(j,k)}(E)$, unde $m(j, k)$ este lungimea cuvîntului $T_{jk} \in K^*$.

Să considerăm gramaticile de tip 2, $G_j = (K, I, E_j; F)$ unde $F = \{E_h \to T_{hk} \mid k = \overline{1, n(h)}, h = \overline{1, n}\}$. Vom arăta că $L(G_j) = D_j$, unde $\mathbf{D} = (D_1, ..., D_n)$ este soluția (II.7.3) a sistemului $\mathbf{E} = \mathbf{H}(\mathbf{E})$. Mai întîi vom dovedi că $D_j \subseteq L(G_j)$. Pentru aceasta este suficient să arătăm că $D_j^{(h)} \subseteq L(G_j)$, $\forall h \in N$.

Procedăm prin inducție asupra lui h. Pentru $h = 0$ fie $p \in D_j^{(0)} = H_j(\varnothing, ..., \varnothing)$. Atunci p este un term al lui $H_j(\mathbf{E})$, $p = T_{jk} \in I^*$. Prin urmare $E_j \to p$ este o regulă de generare din F și p fiind alcătuit numai din simboli terminali, $p \in L(G_j)$, deci $D_j^{(0)} \subseteq L(G_j)$.

Să presupunem că $D_j^{(h)} \subseteq L(G_j)$ și să arătăm că $D_j^{(h+1)} \subseteq L(G_j)$. Fie $p \in D_j^{(h+1)} = H_j(\mathbf{D}^{(h)})$. Există $k \in N$ astfel încît $p \in T_{jk}(\mathbf{D}^{(h)})$ adică $p \in t_{jk1}(\mathbf{D}^{(h)}) ... t_{jkm(j,k)}(\mathbf{D}^{(h)})$.

Putem scrie deci $p = i_1 ... i_{m(j,k)}$ unde

$$i_v = \begin{cases} t_{jkv} \text{ dacă } t_{jkv} \in I^* \\ q_v \in D_{v(v)}^{(h)}, \text{ dacă } t_{jkv} = E_{v(v)}. \end{cases}$$

Ținînd seama de ipoteza inductivă, $D_{v(v)}^{(h)} \subseteq L(G_{v(v)})$, avem, în cazul $i_v = q_v$ deducția $E_{v(v)} \overset{*}{\underset{G_{v(v)}}{\Rightarrow}} i_v$,

Urmează deci că:

$$E_j \underset{G_j}{\Rightarrow} t_{jk1}(E) ... t_{jkm(j,k)}(E) \overset{*}{\underset{G_j}{\Rightarrow}} i_1 ... i_{m(j,k)} = p,$$

adică $E_j \overset{*}{\underset{G_j}{\Rightarrow}} p$ sau $p \in L(G_j)$. Astfel $D_j^{(h+1)} \subseteq L(G_j)$, $\forall j$ și în definitiv $D_j \subseteq L(G_j)$.

Invers, să arătăm că $L(G_j) \subseteq D_j$, $\forall j$.

Să observăm, pentru aceasta că va fi suficient să dovedim că dacă $r_1(\mathbf{E})$, $r_2(\mathbf{E}) \in K^*$ și $r_1(\mathbf{E}) \overset{*}{\underset{G_j}{\Rightarrow}} r_2(\mathbf{E})$ atunci $r_1(\mathbf{D}) \supseteq r_2(\mathbf{D})$. Aceasta, deoarece dacă

[1] Denumirea engleză pentru aceste limbaje este „ALGOL-like languages".

[2] Reamintim că, pentru simplitate, am convenit să notăm cu aceeași literă termul și limbajul generat de el.

$p \in L(G_j)$ avem $E_j \overset{*}{\underset{G_j}{\Rightarrow}} p$ și $D_j = H_j(\mathbf{D}) \supseteq p(\mathbf{D}) = \{p\}$, deoarece p fiind un cuvînt din alfabetul I el nu depinde efectiv de $\mathbf{D}$. Astfel $p \in D_j$ și deci $L(G_j) \subseteq D_j$.

Fie deci $r_1(\mathbf{E}), r_2(\mathbf{E}) \in K^*$ și $r_1(\mathbf{E}) \overset{*}{\underset{G_j}{\Rightarrow}} r_2(\mathbf{E})$. Să presupunem că în gramatica G_j avem deducția :

$r_1(\mathbf{E}) = u_0(\mathbf{E}) \Rightarrow u_1(\mathbf{E}) \Rightarrow \ldots \Rightarrow u_h(\mathbf{E}) = r_2(\mathbf{E})$.

Pentru a arăta că $r_1(\mathbf{D}) \supseteq r_2(\mathbf{D})$ vom proceda prin inducție după h.

Fie, mai întîi $h = 1$. Atunci $r_1(\mathbf{E}) \Rightarrow r_2(\mathbf{E})$ și deci

$$r_1(\mathbf{E}) = v_1(\mathbf{E})\, v_3(\mathbf{E})\, v_2(\mathbf{E}),\ r_2(\mathbf{E}) = v_1(\mathbf{E})\, v_4(\mathbf{E})\, v_2(\mathbf{E}) \text{ și}$$

$(v_3 \to v_4) \in F$. Din definiția lui F, există $j, k \in N$ atsfel încît $v_3(\mathbf{E}) = E_j$, $v_4 = T_{j1}(\mathbf{E})$. Cum D_j este soluția sistemului (II.7.2) avem

$D_j = H_j(\mathbf{D}) = T_j(\mathbf{D}) \cup \bigcup_{k \neq l} T_{jk}(\mathbf{D})$. Apoi $v_3(\mathbf{D}) = D_j = T_{j1}(D) \cup \bigcup_{k \neq l} T_{jk}(\mathbf{D}) =$

$= v_4(\mathbf{D}) \cup \bigcup_{k \neq l} T_{jk}(\mathbf{D}) \supseteq v_4(\mathbf{D})$. De aici obținem

$$r_1(\mathbf{D}) = v_1(\mathbf{D})\, v_3(\mathbf{D})\, v_2(\mathbf{D}) \supseteq v_1(\mathbf{D})\, v_4(\mathbf{D})\, v_2(\mathbf{D}) = r_2(\mathbf{D}).$$

Presupunem că afirmația este adevărată pentru $h \leqslant k$ și fie $r_1(\mathbf{E})$ și $r_2(\mathbf{E})$ două cuvinte astfel încît :

$$r_1(\mathbf{E}) = u_0(\mathbf{E}) \Rightarrow u_1(\mathbf{E}) \Rightarrow \ldots \Rightarrow u_{k+1}(\mathbf{E}) = r_2(\mathbf{E}).$$

Din ipoteza inductivă avem mai întîi $r_1(\mathbf{D}) \supseteq u_k(\mathbf{D})$ și apoi $u_k(\mathbf{D}) \supseteq r_2(\mathbf{D})$, deci $r_1(\mathbf{D}) \supseteq r_2(\mathbf{D})$. ■

Teorema II.7.4. *Limbajele de tip* 2 *sînt limbaje de tip* ALGOL.

Demonstrație. Fie $G = (I_N, I_T, x_0; F)$ o gramatică de tip 2. Fără a restrînge generalitatea putem presupune că $I_N = I_T \cup \{x_0, \ldots, x_n\}$ deoarece adăugarea simbolilor terminali la cei neterminali nu influențează limbajul generat de gramatică. Avem $F = \bigcup_{j=0}^{n} F_j$, unde $F_j = \{x_j \to r_k^{(j)} \mid k = \overline{1, \tau(j)}\}$.

Să considerăm funcțiile standard $H_j(x) = \bigcup_{k=1}^{\tau(j)} \{r_k^{(j)}\}$ unde $x = (x_0, x_1, \ldots, x_n)$.

Gramtica G_0 construită ca în teorema precedentă coincide cu G; ca atare limbajul $L(G)$ va fi prima componentă a soluției de tip (II.7.3) a sistemului $\mathbf{x} = \mathbf{H}(\mathbf{x})$ unde $\mathbf{H} = (H_0, \ldots, H_n)$.

Conchidem că $L(G)$ este un limbaj de tip ALGOL. ■

Observația II.7.1. Fie E un limbaj de tip 2 generat de gramatica $G = (I_N, I_T, x_0; F)$ și $F = \bigcup_{j=0}^{n} F_j$ cu $F_j = \{x_j \to r_k^{(j)} \mid k = \overline{1, \tau(j)}\} = \bigcup_{i \in I_I} F_{ji} \cup \bigcup_{x \in I_N} F_{jx}$, unde $F_{ji} = \{x_j \to ir \mid r \in I^*\}$, $F_{jx} = \{x_j \to xr \mid r \in I^*\}$, Vom face presupunerea că dacă $(x \to e) \in F$ atunci $x = x_0$, fără a micșora generalitatea.

În acest caz sistemul de ecuație $\mathbf{x} = \mathbf{H}(\mathbf{x})$ din demonstrația teoremei precedente devine :

$$x_j = \bigcup_{i_k \in I} i_k \{r_{jk}\} \cup \bigcup_{x_h \in I_N} x_h r_{jh} \cup \varepsilon_j, \; j = \overline{0, n},$$

$$\varepsilon_j = \begin{cases} \varnothing, \text{ dacă } j \neq 0 \text{ sau } j = 0 \text{ și } e \notin E \\ \{e\}, \text{ dacă } j = 0 \text{ și } e \in E \end{cases}$$

Teorema II.7.5. *Clasa limbajelor de tip* ALGOL *coincide cu clasa limbajelor de tip* 2.

Demonstrație. Rezultă din teoremele II.7.3 și II.7.4. ■

Exemplul II.7.1. Fie $I_T = \{i_1, i_2, i_3\}$ și $H(x_0, x_1) = (\{i_1 x_0\} \cup \{x_1 i_1 i_2 x_0\} \cup \{i_3\}, \{x_0 x_1\} \cup \{i_1 x_0\} \cup \{i_1 i_3\})$.

Să considerăm gramatica $G = (\{x_0, x_1\} \cup \{i_1, i_2, i_3\}, \{i_1, i_2, i_3\}, x_0; F)$ unde

$$F = x_0 \to i_1 x_0, \; x_0 \to x_1 i_1 i_2 x_0, \; x_0 \to i_3, \; x_1 \to x_0 x_1, \; x_1 \to i_1 x_0, \; x_1 \to i_1 i_3\}.$$

Atunci $L(G)$ este prima componentă a soluției sistemului:

$$\text{(II.7.4)} \qquad \begin{cases} E_0 = i_1 E_0 \cup E_1 i_1 i_2 E_0 \cup \{i_3\} \\ E_1 = E_0 E_1 \cup i_1 E_0 \cup \{i_1 i_3\}. \end{cases}$$

Soluția (II.7.3) a sistemului (II.7.4) poate fi construită astfel:

$$\mathbf{D}^{(0)} = \mathbf{H}(\varnothing, \varnothing) = (\{i_3\}, \{i_1 i_3\}),$$
$$\mathbf{D}^{(1)} = \mathbf{H}(\mathbf{D}^{(0)}) = (\{i_1 i_3, i_1 i_3 i_1 i_2 i_3, i_3\}, \{i_3 i_1 i_2, i_1 i_3\}),$$

și $\mathbf{D} = \bigcup_{j=0}^{\infty} \mathbf{D}^{(j)}$.

Teorema II.7.6. *Soluția* (II.7.3) *a sistemului* (II.7.2) *este primul element al mulțimii soluțiilor sistemului.*

Demonstrație. Conform teoremei II.7.2 mulțimea soluțiilor sistemului (II.7.2) este nevidă. Fie $\mathbf{E} = (E_1, \ldots, E_n)$ o soluție oarecare și $\mathbf{D} = (D_1, \ldots, D_n)$ soluția de tip (II.7.3).

Vom demonstra prin inducție după j, că $\mathbf{D}^{(j)} \subseteq \mathbf{E}$, $\forall j \in N$, de unde va rezulta că $\mathbf{D} = \bigcup_{j=0}^{\infty} \mathbf{D}^{(j)} \subseteq \mathbf{E}$.

Fie $j = 0$. Deoarece $\varnothing \subseteq \mathbf{E}$ urmează $\mathbf{D}^{(0)} = \mathbf{H}(\varnothing) \subseteq \mathbf{H}(\mathbf{E}) = \mathbf{E}$ (v. teorema (II.7.1)). Să presupunem că $\mathbf{D}^{(j)} \subseteq \mathbf{E}$. Avem

$$\mathbf{D}^{(j+1)} = (\mathbf{D}^{(j)}) \subseteq \mathbf{H}(\mathbf{E}) = \mathbf{E}. \; ■$$

§.II.8. AUTOMATE PUSHDOWN (AUTOMATE CU PILE DE MEMORIE)

Ne vom ocupa în acest paragraf de o clasă de dispozitive care joacă pentru limbajele de tip 2 același rol pe care-l joacă automatele finite față de limbajele de tip 3, și anume de automatele pushdown.

Spre deosebire de automatele finite, *automatele pushdown* posedă atît memorie internă (a cărei activitate se exprimă prin evoluţia stărilor interne ale automatului) cît şi o memorie externă de un tip special, memoria pushdown.

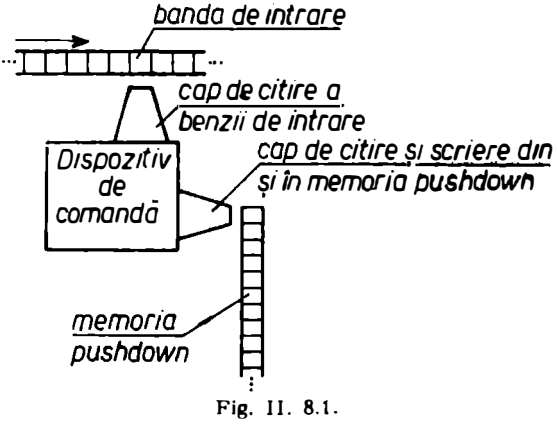

Fig. II. 8.1.

O memorie de tip pushdown[1] este o bandă nemărginită în jos alcătuită din celule în care se poate memora cîte un simbol dintr-un anume alfabet. Introducerea şi extragerea informaţiei în memoria pushdown se face numai prin intermediul primei celule. Astfel, introducînd un simbol în prima celulă conţinutul tuturor celorlalte se deplasează în jos cu o celulă, iar extrăgînd conţinutul primei celule se produce o deplasare în sus cu o celulă. Prin urmare, ultimul simbol introdus este în mod necesar primul simbol extras şi primul simbol introdus este ultimul extras.

Un *automat pushdown* lucrează în mod discret, în tacte şi conţine trei părţi :

– O bandă de intrare împărţită în celule care pot conţine cîte un simbol dintr-un alfabet de intrare I, numărul celulelor ocupate fiind finit.

– Un dispozitiv de comandă, care conţine memoria internă, capabil să se afle într-o anume stare dintr-o mulţime finită de stări S. Aceasta posedă un „cap de citire" a benzii de intrare şi un alt cap pentru citirea şi scrierea informaţiilor din şi în memoria pushdown.

– O memorie pushdown care conţine simboli dintr-un alfabet Z (v. fig. II.8.1).

Funcţionarea unui astfel de automat într-un anume tact depinde de simbolul citit de pe banda de intrare, de starea internă a dispozitivului de comandă şi de conţinutul primei celule din memoria pushdown.

Banda de intrare se poate deplasa în fiecare tact cu cîte o celulă la dreapta trecînd prin faţa capului de citire sau poate rămîne pe loc în care caz se consideră că dispozitivul citeşte cuvîntul nul. În citirea unui simbol de intrare din $I \cup \{e\}$ dispozitivul de comandă efectuează două acţiuni funcţie de starea în care se află, de simbolul de intrare citit şi de conţinutul primei celule a memoriei pushdown :

– trece într-o nouă stare internă

– înscrie în memoria pushdown înlocuind primul simbol cu un cuvînt $r \in Z^*$ (care poate fi eventual cuvîntul nul, în care caz se produce de fapt

[1] Denumirea de „pushdown" descrie sugestiv funcţionarea memoriei. Ea provine de la cuvintele englezeşti to push = a împinge şi down = jos.

ştergerea conţinutului primei celule). Bineînţeles că la introducerea unui cuvînt de lungime $k > 0$ se produce deplasarea în jos a conţinutului memoriei pushdown cu k celule.

Începerea activităţii automatului se face dintr-o stare, starea iniţială s_0 şi un anumit conţinut $z_0 \in Z$ al primei celule din memoria pushdown numit *simbol de start.*

Recunoaşterea unui cuvînt de pe banda de intrare are loc cînd, după citirea ultimului său simbol memoria pushdown s-a golit complet. Formalizarea acestei noţiuni se face în :

Definiţia II.8.1. Un *automat pushdown* este un 6-uplu $APD = (I, S, Z, f; s_0, z_0)$ unde I, S, Z sînt mulţimi finite nevide reprezentînd respectiv alfabetul de intrare, mulţimea de stări interne şi alfabetul memoriei pushdown al automatului, $f: S \times (I \cup \{e\}) \times Z \to \mathcal{P}_f(S \times Z^*)$ numită funcţia de tranziţie, $s_0 \in S$ este starea internă iniţială, iar $z_0 \in Z$ simbolul de start.

Prin *stare globală* a automatului pushdown înţelegem un cuplu $(s, q) \in S \times Z^*$.

Observaţia II.8.1. Automatul pushdown descris în definiţia II.8.1 funcţionează în mod *nedeterminist* în sensul că în fiecare tact el are posibilitatea mai multor alegeri pentru noua stare internă şi pentru cuvîntul înscris în memoria pushdown. Avem totdeauna $|f(s, i, z)| < \infty$, iar dacă $|f(s, i, z)| = 1$, $\forall s \in S$, $i \in I \cup \{e\}$, $z \in Z$ automatul funcţionează în mod *determinist.* Dacă $f(s, i, z) = \varnothing$ atunci simbolul i este inaplicabil la intrarea automatului cînd starea internă este s şi z se află în capul memoriei pushdown.

Definiţia II.8.2. O *configuraţie instantanee* a unui automat pushdown $APD = (I, S, Z, f; s_0, x_0)$ este un triplet $\delta = (s, p, q) \in S \times I^* \times Z^*$.

Definiţia II.8.3. *Configuraţia instantanee* $\delta_1 = (s, ip, zq)$, unde $i \in I \cup \{e\}$, $z \in Z$ precede imediat configuraţia instantanee $\delta_2 = (s', p, q_1 q)$ dacă şi numai dacă $(s', q_1) \in f(s, i, z)$, ceea ce se va nota cu $\delta_1 \vdash \delta_2$.

Configuraţia instantanee δ *precede configuraţia instantanee* δ', dacă există un şir finit de configuraţii instantanee $\delta = \delta_0, \delta_1, \ldots, \delta_n = \delta'$ cu $\delta_j \vdash \delta_{j+1}$, $j = \overline{0, n-1}$.

Vom nota aceasta cu $\delta \overset{*}{\vdash} \delta'$. Şirul $\delta = \delta_0 \vdash \delta_1 \vdash \ldots \vdash \delta_n = \delta'$ îl vom numi *calcul al configuraţiei* δ' pornind de la configuraţia δ, iar n va fi numit *lungimea calculului.*

Observaţia II.8.2. Configuraţia instantanee (s, p, e) nu poate precede nici o altă configuraţie instantanee deoarece funcţia f nu este definită pentru tripletele de forma (s, p, e). Din acest motiv, la începutul activităţii în memoria pushdown se află totdeauna simbolul de start $z_0 \in Z$.

Observaţia II.8.3. Configuraţia instantanee (s, p, q) reprezintă faptul că automatul se găseşte în starea internă s, că în tactele următoare se va prelucra cuvîntul p de pe banda de intrare şi că în memoria pushdown se află cuvîntul q scris de jos în sus, simbolul cel mai de la dreapta al cuvîntului aflîndu-se în prima celulă (din „vîrf").

Relația $\delta_1 \vdash \delta_2$ indică faptul că automatul se află în starea s, că z este simbolul din „vîrful" memoriei pushdown și că i este simbolul de intrare care trebuie să fie prelucrat sau că $i = e$ și singura evoluție posibilă prelucrîndu-l pe i este trecerea în starea s' și înlocuirea lui z prin cuvîntul q_1.

Relația $(s, i_1, ..., i_k\ p, q) \overset{\bullet}{\vdash} (s', p, q')$ indică faptul că automatul se află în starea s, q este conținutul memoriei pushdown, că $i_1 ... i_k\ p$ este partea ce urmează să fie prelucrată la intrare și după *cel puțin k* tacte (deoarece în unele tacte banda de intrare nu se deplasează) automatul trece în starea s', conținutul memoriei pushdown este q' și rămîne de prelucrat cuvîntul p de pe banda de intrare.

Automatele pushdown pot servi la recunoașterea unor limbaje așa cum rezultă din :

Definiția II.8.4. Limbajul $L(APD)$ recunoscut de automatul $APD = (I, S, Z, f; s_0, z_0)$ este

$$L(APD) = \{p \mid \exists s \in S,\ (s_0, p, z_0) \overset{\bullet}{\vdash} (s, e, e)\}.$$

Vom ilustra noțiunea de automat pushdown în :

Exemplul II.8.1. Fie $APD = (\{i_1, i_2\}, \{s_0\}, \{i_1, i_2, z_0\}, f; s_0; z_0)$ unde f este dată de următorul tablou :

$f(s_0, i, z)$	i_1	i_2	z_0
i_1	$\{(s_0, e)\}$	$\varnothing$	$\varnothing$
i_2	$\varnothing$	$\{(s_0, e)\}$	$\varnothing$
e	$\varnothing$	$\varnothing$	$\{(s_0, i_1 z_0 i_2), (s_0, i_1 i_2)\}$

Introducînd cuvîntul $p = i_1 i_1 i_1 i_2 i_2 i_2$ putem obține, în funcție de alegerile pe care le face automatul mai multe șiruri de precedențe imediate printre care avem :

$$(s_0, i_1 i_1 i_1 i_2 i_2 i_2, z_0) \vdash (s_0, i_1 i_1 i_1 i_2 i_2 i_2, i_1 z_0 i_2) \vdash$$
$$\vdash (s_0, i_1 i_1 i_2 i_2 i_2, z_0 i_2) \vdash (s_0, i_1 i_1 i_2 i_2 i_2, i_1 z_0 i_2 i_2) \vdash$$
$$\vdash (s_0, i_1 i_2 i_2 i_2, z_0 i_2 i_2) \vdash (s_0, i_1 i_2 i_2 i_2, i_1 i_2 i_2 i_2) \vdash$$
$$\vdash (s_0; i_2 i_2 i_2, i_2 i_2 i_2) \vdash (s_0, i_2 i_2, i_2 i_2) \vdash (s_0; i_2, i_2) \vdash (s_0, e, e),$$

sau, de pildă, șirul :

$$(s_0, i_1 i_1 i_1 i_2 i_2 i_2, z_0) \vdash (s_0, i_1 i_1 i_1 i_2 i_2 i_2, i_1 i_2) \vdash$$
$$\vdash (s_0, i_1 i_1 i_2 i_2 i_2, i_2),$$

care nu mai poate continua simbolul i_1 nefiind aplicabil atunci cînd automatul se găsește în starea s_0 avînd în capătul memoriei pushdown simbolul i_2. Cu toate acestea, existența primului șir de precedențe imediate, adică

$$(s_0, i_1 i_1 i_1 i_2 i_2 i_2, z_0) \overset{\bullet}{\vdash} (s_0, e, e)$$

asigură că $i_1 i_1 i_1 i_2 i_2 i_2 \in L(APD)$.

Se verifică imediat că :

$$L(APD) = \{\underbrace{i_1 \ldots i_1}_{n} \underbrace{i_2 \ldots i_2}_{n} \mid n \in N^*\}.$$

Pe lîngă definiția II.8.4 a limbajelor recunoscute de automate pushdown (în care recunoașterea unui cuvînt se face prin golirea memoriei pushdown) vom prezenta un alt punct de vedere apropiat de reprezentarea limbajelor regulate în automate finite descris în :

Definiția II.8.5. *Limbajul reprezentat într-un automat pushdown* $APD = (I, S, Z, f; s_0, z_0)$ *prin mulțimea de stări finale* $S_1 \subseteq S$ este :

$$L(APD)(S_1) = \{p \mid \exists q \in Z^*, s \in S_1, (s_0, p, z_0) \overset{\bullet}{\vdash} (s, e, q)\}.$$

Legătura dintre definițiile II.8.4 și II.8.5 este dată în :

Teorema II.8.1. *Un limbaj L este recunoscut de un automat pushdown dacă și numai dacă este reprezentat într-un automat pushdown de o mulțime de stări finale.*

Demonstrație. Fie $APD = (I, S, Z, f; s_0, z_0)$ un automat pushdown, mulțimea de stări finale $S_1 \subseteq S$ și

$$L(APD)(S_1) = \{p \mid \exists q \in Z^*, s \in S_1, (s_0, p, z_0) \overset{\bullet}{\vdash} (s, e, q)\}.$$

Să construim un automat pushdown în care acest limbaj să fie recunoscut. Pentru aceasta fie :

$$A' = (I, S \cup \{s', s_0'\}, Z \cup \{z'\}, f'; s_0', z'),$$

unde funcția f' este definită prin :

$$f'(s, i, z) = \begin{cases} (s_0, z_0, z'), \text{ dacă } (s, i, z) = (s_0', e, z') \\ (s', e), \ \forall s \in S_1 \cup \{s'\}, i = e, z \in Z \cup \{z'\} \\ f(s, i, z), \ \forall s \in S, i \in I \cup \{e\}, z \in Z \\ \varnothing, \text{ în orice alte situații.} \end{cases}$$

Introducerea simbolului z' pentru automatul A' se justifică prin faptul că pot exista cuvinte în $I^* \setminus L(APD)(S_1)$ care să realizeze în automatul APD golirea memoriei pushdown. Acest lucru nu mai este posibil în automatul A' din cauză că simbolul z' este prezent în ultima celulă a memoriei pushdown. Egalitatea $f'(s_0', e, z') = (s_0, z_0 z')$ exprimă faptul că automatul A' își începe funcționarea prin trecerea din starea s_0' în starea s_0 și prin pla-

sarea cuvîntului $z_0\, z'$ în memoria pushdown. Dacă APD intră într-o stare finală sub influenţa unui cuvînt p, sub influenţa aceluiaşi cuvînt automatul A' poate trece în starea s' şi apoi să efectueze ştergerea memoriei pushdown inclusiv a simbolului z'.

Fie $p \in L(APD)(S_1)$. Atunci există un calcul

$$(s_0, p, z_0) \overset{*}{\underset{APD}{\vdash}} (s, e, q),\ s \in S_1,$$

pe care-l folosim pentru a pune în evidenţă un calcul în automatul A' care să conducă la golirea memoriei pushdown.

În baza definiţiei funcţiei f' avem:

$$(s_0', p, z') \underset{A'}{\vdash} (s_0, p, z_0\, z') \overset{*}{\underset{A'}{\vdash}} (s, e, q\, z') \overset{*}{\underset{A'}{\vdash}} (s', e, e).$$

Astfel $p \in L(A')$.

Invers, fie $p \in L(A')$. Există un calcul

$$(s_0', p, z') \overset{*}{\underset{A'}{\vdash}} (s, e, e). \tag{II.8.1}$$

Prima configuraţie în care intră automatul A' este în mod necesar (s_0, p, z_0, z'). Astfel, calculul (II.8.1) se mai scrie

$$(s_0', p, z') \vdash (s_0, p, z_0\, z') \overset{*}{\underset{A'}{\vdash}} (s, e, e).$$

Din definiţia funcţiei f' rezultă că configuraţiile care se succed sînt cele ale automatului APD cu excepţia prezenţei simbolului z' din celula cea mai de jos a memoriei pushdown. După prelucrarea cuvîntului p configuraţia automatului A' va fi $(\bar{s}, e, qz')$, unde $q \in Z$ şi $\bar{s} \in S$.

Simbolul z' nu poate fi şters, aşa cum se remarcă din definiţia funcţiei f' decît plecînd de la o stare din S_1 sau din starea s'. Avînd $\bar{s} \in S$, automatul APD nu poate trece în starea s' şi cum ştergerea simbolului z' are loc în orice caz rezultă că automatul APD trebuie să treacă în mod obligatoriu într-o stare finală (din mulţimea S_1) sub influenţa cuvîntului p. Aşadar $\bar{s} \in S_1$ şi avem $(s_0, p, z_0) \overset{*}{\underset{APD}{\vdash}} (\bar{s}, e, q)$, adică $p \in L(APD)(S_1)$. Astfel $L(A') = L(APD)(S_1)$.

Reciproc, fie $APD = (I, S, Z, f; s_0, z_0)$ un automat pushdown şi $L(APD)$ limbajul recunoscut de acest automat. Vom construi un automat pushdown în care acest limbaj să fie reprezentat printr-o mulţime de stări finale. Fie deci $A' = (I, S \cup \{s_0', s_f\}, Z \cup \{z'\}, f'; s_0', z_0)$ un automat pushdown în care f' este dat prin:

$$f'(s, i, z) = \begin{cases} (s_0, z_0\, z'), \text{ dacă } (s, i, z) = (s_0', e, z') \\ f(s, i, z), \ \forall s \in S, \ i \in I \cup \{e\}, \ z \in Z \\ (s_f, e), \ \forall s \in S, \ i = e, \ z = z' \\ \varnothing, \text{ în orice alte situații.} \end{cases}$$

Vom arăta că $L(APD) = L(A')(\{s_f\})$, adică în automatul A' limbajul $L(APD)$ este reprezentat prin starea finală $\{s_f\}$.

Fie $p \in L(APD)$. Există un calcul :

$$(s_0, p, z_0) \overset{*}{\underset{APD}{\vdash}} (s, e, e).$$

Să considerăm configurația inițială (s_0', p, z') în automatul A'. Configurația care o succede imediat pe aceasta nu poate fi decît $(s_0, p, z_0\, z')$ așa cum rezultă din definiția funcției f'. În continuare configurațiile care se succed sînt cele ale automatului APD cu excepția prezenței simbolului z' în celula cea mai de jos a memoriei pushdown a automatului A'. Atunci, după prelucrarea cuvîntului p putem obține în A' configurația (s, e, z') și automatul A' poate trece în configurația (s_f, e, e), deci $p \in L(A')(\{s_f\})$.

Invers, fie $p \in L(A')(\{s_f\})$. Există un calcul

$$(s_0', p, z') \overset{*}{\vdash} (s_f, e, e), \tag{II.8.2}$$

deoarece automatul A' nu poate ajunge în starea s_f decît la ștergerea ultimului simbol z' din memorie.

Calculul (II.8.2) poate fi detaliat :

$$(s_0', p, z') \underset{A'}{\vdash} (s_0, p, z_0, z') \overset{*}{\underset{A'}{\vdash}} (\bar{s}, e, z') \underset{A'}{\vdash} (s_f, e, e).$$

Corespunzător calculului $(s_0, p, z_0\, z') \overset{*}{\underset{A'}{\vdash}} (\bar{s}, e, z')$ în automatul APD avem calculul

$$(s_0, p, z_0) \overset{*}{\underset{APD}{\vdash}} (\bar{s}, e, e),$$

deci $p \in L(APD)$. ■

În continuare vom pune în evidență legătura dintre limbajele de tip 2 și limbajele recunoscute de automatele pushdown. Mai precis, vom arăta coincidența celor două clase de limbaje.

Teorema II.8.2. *Dacă L este un limbaj de tip 2 există un automat pushdown în care limbajul L poate fi reprezentat.*

Demonstrație. Să presupunem mai întîi că $e \notin L$. Fie $G = (I_N, I_T, x_0 ; F)$ o gramatică de tip 2, de forma normală S. Greibach (v. teorema II.5.12) care generează limbajul L. Fie $A = (I_T, \{s\}, I_N, f ; s, x_0)$, unde $f(s, i, x) = (s, r)$, dacă $(x \to ir) \in F$ și $r \in I_N^*$ și arbitrar în rest. Vom arăta că $L = L(A)$.

Fie $p = i_1 \dots i_n \in L$. Există deducția $x_0 \overset{\bullet}{\underset{G}{\Rightarrow}} p$, care se poate scrie detaliat: $x_0 \Rightarrow i_1\ r_1 \Rightarrow i_1 i_2\ r_2 \Rightarrow \dots \Rightarrow i_1 \dots i_{n-1}\ r_{n-1} \Rightarrow i_1 \dots i_n = p$[1]), unde $r_j \in I_N^*$, $j = \overline{1, n-1}$ și poate fi considerată, fără restrîngerea generalității ca deducție imediată la stînga (v. teorema II.5.6).

Fie (s, p, x_0) configurația inițială a automatului A. Conform definiției funcției f în automatul A putem avea calculul:

$$(s, i_1 \dots i_n, x_0) \underset{A}{\vdash} (s, i_2 \dots i_n, r_1) \underset{A}{\vdash} (s, i_3 \dots i_n, r_2) \underset{A}{\vdash} \dots$$
$$\dots \underset{A}{\vdash} (s, i_n, r_{n-1}) \underset{A}{\vdash} (s, e, e),$$

deci $i_1 i_2 \dots i_n = p \in L(A)$.

Invers, fie $p \in L(A)$. Există calculul

$$(s, p, x_0) \overset{\bullet}{\underset{A}{\vdash}} (s, e, e).$$

Vom arăta, prin inducție asupra lungimii cuvîntului p că dacă există un calcul $(s, p, r) \overset{\bullet}{\underset{A}{\vdash}} (s, e, e)$ atunci există o deducție $r \overset{\bullet}{\underset{G}{\Rightarrow}} p$.

Fie $l(p) = 1$, adică $p = i \in I_T$. Avem $(s, i, x_0) \underset{A}{\vdash} (s, e, e)$ și corespunzător avem regula de generare $(x_0 \to i) \in F$ și deci $i \in L$.

Presupunem afirmația adevărată pentru cuvinte p cu $l(p) \leqslant n-1$ și fie $q \in I_T^*$ cu $l(q) = n$ și $q = ip$, cu $l(p) = n-1$.

Deoarece $(s, q, x_0) = (s, ip, x_0) \underset{A}{\vdash} (s, p, r) \underset{A^*}{\vdash} (s, e, e)$, unde $(x_0 \to ir) \in F$, în baza ipotezei inductive avem deducția $r \Rightarrow p$, de unde rezultă existența deducției $x_0 \Rightarrow ir \Rightarrow ip = q$, deci $L = L(A)$.

Dacă $e \in L$ considerăm limbajul $L \setminus \{e\}$, atașăm gramaticii sub forma normală S. Greibach regula de generare $x_0 \to e$ și admitem că $f(s, e, x_0) = (s, e)$, adică permitem automatului A să prelucreze și cuvîntul nul efectuînd ștergerea memoriei pushdown. ■

Teorema II.8.3. *Dacă limbajul L este reprezentat într-un automat pushdown, L este un limbaj de tip 2.*

Demonstrație. Fie $APD = (I, S, Z, f; s_0, z_0)$ un automat pushdown și $L = L(APD)$. Vom construi o gramatică de tip 2, G astfel ca $L = L(G)$. Anume, $G = (I_N, I, x_0; F)$, unde $I_N = \{s_0\} \cup \{(s_1, z, s_2) \mid s_1, s_2 \in S, z \in Z\}$, iar F este mulțimea regulilor de generare care se pot forma prin următoarele două procedee:

1) $x_0 \to (s_0, z_0, s)$, $\forall s \in S$

2) $(s, z, s') \to i\ (s_1, z_1, s_2)\ (s_2, z_2, s_3) \dots (s_m, z_m, s_{m+1})$ pentru $\forall s, s_1, \dots, s_{m+1} \in S$ cu $s' = s_{m+1}$, $\forall i \in I \cup \{e\}$, $z, z_1, \dots, z_m \in Z$, astfel ca $(s_1, z_1, z_2, \dots, z_m) \in f(s, i, z)$. Dacă $m = 0$ atunci $s_1 = s'$, $(s', e) \in f(s, i, z)$ și regula de generare devine $(s, z, s') \to i$, unde $i \in I \cup \{e\}$.

[1]) Remarcăm că ultima regulă de generare întrebuințată este în mod necesar de forma $r_{n-1} = x \to i_n$.

Să presupunem că $p \in L(APD)$, adică există calculul

(II.8.3) $$(s_0, p, z_0) \overset{\bullet}{\underset{APD}{\vdash}} (s, e, e).$$

Vom arăta că $x_0 \overset{*}{\underset{G}{\Rightarrow}} p$, adică $p \in L(G)$.

Pentru aceasta, vom demonstra prin inducție după lungimea calculului, că dacă există calculul :

(II.8.4) $$(\bar{s}, p, z) \overset{\bullet}{\underset{APD}{\vdash}} (s, e, e),$$

atunci există deducția $(\bar{s}, z, s) \overset{\bullet}{\underset{G}{\Rightarrow}} p$.

Fie un calcul (II.8.4) de lungime $n = 1$. Atunci $l(p) \leqslant 1$ și $p = i \in I \cup \{e\}$. Astfel, $(\bar{s}, i, z) \vdash (s, e, e)$, $(s, e) \in f(\bar{s}, i, z)$ și prin urmare $((\bar{s}, z, s) \to i) \in F$ și deci există deducția $(\bar{s}, z, s) \Rightarrow i$.

Să presupunem că afirmația este valabilă pentru calcule (II.8.4) de lungime cel mult $n - 1$ și fie $(\bar{s}, q, z) \overset{*}{\underset{APD}{\vdash}} (s, e, e)$ un calcul de lungime n. Avem $l(q) \leqslant n$ și $q = ip$, cu $l(p) \leqslant n - 1$. Calculul de mai sus poate fi scris și : $(\bar{s}, q, z) = (\bar{s}, ip, z) \underset{APD}{\vdash} (s_1, p, r) = (s_1, p_1 \ldots p_m, z_1 \ldots z_m) \overset{\bullet}{\underset{APD}{\vdash}} (s, e, e)$, cu $m \geqslant 1$ astfel încît $\forall j = \overline{1, m}$ are loc calculul

$$(s_j, p_j, z_j) \overset{\bullet}{\underset{APD}{\vdash}} (s_{j+1}, e, e).$$

având lungimea cel mult $n - 1$. Aici $s_{m+1} = s$.

Cuvîntul p a fost împărțit în subcuvintele $p_1, p_2, \ldots, p_m$ astfel încît în urma prelucrării cuvîntului p_j să se realizeze ștergerea simbolului z_j din memoria pushdown, $j = \overline{1, m}$. În acest caz, conform ipotezei inductive există deducțiile :

$$(s_j, z_j, s_{j+1}) \overset{\bullet}{\underset{G}{\Rightarrow}} p_j, \quad j = \overline{1, m}, \; s_{m+1} = s.$$

Cum în gramatica G avem regula de generare :

$$(\bar{s}, z, s) \to i\,(s, z_1, s_2)(s_2, z_2, s_3) \ldots (s_m, z_m, s_{m+1}).$$

(deoarece $f(\bar{s}, i, z) \ni (s_1, z_1, z_2 \ldots z_m)$) rezultă că :

$$(\bar{s}, z, s) \overset{\bullet}{\underset{G}{\Rightarrow}} i\, p_1 \ldots p_m = q.$$

Dacă $p \in L(APD)$ atunci există calculul :

$$(s_0 , p, z_0) \overset{\bullet}{\underset{APD}{\vdash}} (s, e, e),$$

de unde, în gramatica G avem deducția :

$$x_0 \underset{G}{\Rightarrow} (s_0 , z_0 , p) \overset{\bullet}{\underset{G}{\Rightarrow}} p,$$

adică $p \in L(G)$.

Să arătăm acum că $L(G) \subseteq L(APD)$. În acest scop vom demonstra că dacă are loc deducția imediată la stînga :

$$(s_0 , z_0 , s) \overset{\bullet}{\underset{G}{\Rightarrow}} p\,(s_\alpha , z_\alpha , s_\beta) \dots (s_\mu , z_\mu , s), \tag{II.8.5}$$

atunci există calculul :

$$(s_0 , p, z_0) \overset{\bullet}{\underset{APD}{\vdash}} (s, e, z_\alpha \dots z_\mu),$$

procedînd prin inducție asupra lungimii h a deducției (II.8.5). Aici $p \in I^*$.

Pentru $h = 1$ cuvîntul p este obligatoriu de forma $p = i \in I \cup \{e\}$, deducția (II.8.5) revine la :

$$(s_0 , z_0 , s) \Rightarrow i\,(s_\alpha , z_\alpha , s_\beta) \dots (s_\mu , z_\mu , s),$$

și $(s_\alpha , z_\alpha \dots z_\mu) \in f(s_0 , i, z_0)$. Avem corespunzător calculul :

$$(s_0 , i, z_0) \underset{APD}{\vdash} (s, e, z_\alpha \dots z_\mu).$$

Presupunem afirmația valabilă pentru $h \leqslant n - 1$ și fie $(s_0 , z_0 , s) \overset{\bullet}{\underset{G}{\Rightarrow}}$ $\overset{\bullet}{\underset{G}{\Rightarrow}} q\,(s_\nu , z_\nu , s_\gamma) \dots (s_\mu , z_\mu , s)$ o deducție de lungime n. Această deducție poate fi detaliată sub forma :

$$(s_0 , z_0 , s) \overset{\bullet}{\underset{G}{\Rightarrow}} p\,(s_\alpha , z_\alpha , s_\beta) \dots (s_\mu , z_\mu , s) \Rightarrow$$

$$\Rightarrow pi\,(s_\nu , z_\nu , s_\gamma) \dots (s_\delta , z_\delta , s_\beta) \dots (s_\mu , z_\mu , s),$$

unde $q = pi$, ultima regulă de generare folosită fiind $(s_\alpha , z_\alpha , s_\beta) \to i\,(s_\nu , z_\nu , s_\gamma) \dots$ $\dots (s_\delta , z_\delta , s_\beta)$ și deci $f(s_\alpha , i, z_\alpha) \ni (s_\nu , z_\nu \dots z_\delta)$.

Pentru cuvîntul p există deci deducția :

$$(s_0 , z_0 , p) \overset{\bullet}{\Rightarrow} p\,(s_\alpha , z_\alpha , s_\beta) \dots (s_\mu , z_\mu , s)$$

de lungime cel mult $n - 1$ și, conform ipotezei inductive există calculul :

$$(s_0, p, z_0) \overset{\bullet}{\vdash} (s_0, e, z_\alpha \dots z_\mu).$$

Ca atare, plecînd de la configurația $(s_0, q, z_0) = (s_0, pi, z_0)$ obținem calculul :

$(s_0, pi, z_0) \overset{\bullet}{\vdash} (s_\alpha, i, z_\alpha \dots z_\mu) \vdash (s_\nu, e, z_\nu \dots z_\delta \dots z_\mu)$, ceea ce ne-am propus să dovedim.

Fie acum $p \in L(G)$. În baza teoremei II.5.6 există o deducție imediată la stînga :

$$x_0 \Rightarrow (s_0, z_0, s) \underset{G}{\overset{\bullet}{\Rightarrow}} p\,(s_\alpha, z_\alpha, s_\beta) \dots (s_\mu, z_\mu, s) \underset{G}{\overset{\bullet}{\Rightarrow}} p,$$

ultimele reguli de generare aplicate fiind $(s_\alpha, z_\alpha, s_\beta) \Rightarrow e, \dots, (s_\mu, z_\mu, s) \to e$.

Conform celor arătate mai sus există, pe de o parte un calcul :

$$(s_0, p, z_0) \underset{APD}{\overset{\bullet}{\vdash}} (s_\alpha, e, z_\alpha \dots z_\mu), \tag{II.8.6}$$

iar pe de altă parte, din definiția gramaticii G avem :

$$(s_\beta, e) \in f(s_\alpha, e, z_\alpha),$$

.

.

$$(s, e) \in f(s_\mu, e, z_\mu).$$

Astfel calculul (II.8.6) poate fi continuat, obținîndu-se calculul :

$$(s_0, p, z_0) \underset{APD}{\overset{\bullet}{\vdash}} (s_\alpha, e, z_\alpha \dots z_\mu) \underset{APD}{\overset{\bullet}{\vdash}} (s, e, e),$$

ceea ce arată că $p \in L(APD)$. ■

Teorema II.8.4. *Următoarele afirmații sînt echivalente* :

1) *L este un limbaj de tip 2.*

2) *L este recunoscut de un automat pushdown.*

3) *L este reprezentat într-un automat pushdown printr-o mulțime de stări finale.*

Demonstrație. Rezultă din teoremele II.8.1 – II.8.3. ■

§ II. 9. AUTOMATE DE TIP LETICEVSKI

Vom prezenta în acest paragraf o clasă de automate deterministe infinite care permit de asemenea recunoașterea limbajelor de tip 2.

Această clasă de automate a fost introdusă de A. A. Leticevski în anul 1965 [1].

Definiţia II.9.1. *Un automat de tip Leticevski AL* este un 4-uplu :

$$AL = (I, \mathcal{P}_f(Z^*), F ; L_0),$$

unde I este un alfabet finit, $\mathcal{P}_f(Z^*)$ este mulţimea stărilor automatului AL (constituită din mulţimea limbajelor finite pe alfabetul finit Z) şi $F : \mathcal{P}_f(Z^*) \times I \to \mathcal{P}_f(Z^*)$ este funcţia de tranziţie a automatului AL care satisface condiţiile :

1) $F(\varnothing, i) = F(\{e\}, i) = \varnothing, \quad \forall i \in I,$

2) $F(zp, i) = F(z, i)\, p$, pentru $\forall i \in I,\ z \in Z,\ p \in Z^*,$

3) $F(L_1 \cup L_2, i) = F(L_1, i) \cup F(L_2, i), \quad \forall i \in I,\ L_1, L_2 \in \mathcal{P}_f(Z^*),$

4) Pentru $p \in F(z, i)$ avem $l(p) \leqslant 2$.

Observaţia II.9.1. În baza condiţiilor 1) – 3) funcţia F este unic determinată prin specificarea mulţimii finite de limbaje $\{F(z, i) \mid z \in Z,\ i \in I\}$.

Condiţia 2) este asemănătoare descrierii funcţionării memoriei pushdown, deoarece sub influenţa simbolului de intrare i este prelucrat numai simbolul z, începutul cuvîntului zp.

Exemplul II.9.1. Fie $AL = (\{i_1, i_2\}, \mathcal{P}_f(Z^*), F ; L_0)$ unde $Z = \{z_0, z_1\}$, $L_0 = \{z_0, z_1\}$, iar F este determinat de tabelul :

F	z_0	z_1
i_1	$\{z_0, z_0z_1\}$	$\{z_1, z_1z_0\}$
i_2	$\{z_1\}$	$\{z_0\}$

Observaţia II.9.2. Funcţia $F : \mathcal{P}_f(Z^*) \times I \to \mathcal{P}_f(Z^*)$ poate fi extinsă la $\mathcal{P}_f(Z^*) \times I$ punînd :

$$F(L, e) = L_0 \quad \forall L \in \mathcal{P}_f(Z^*),$$

$$F(L, pi) = F(L, p), i), \quad \forall p \in I^*,\ i \in I,\ \forall L \in \mathcal{P}_f(Z^*).$$

Definiţia II.9.2. *Limbajul $L(AL)$ reprezentat în automatul AL* este :

$$L(AL) = \{p \mid F(s_0, p) \in \mathcal{P}_f^e(Z^*)\},$$

unde am notat prin $\mathcal{P}_f^e(Z^*)$ familia de limbaje finite asupra lui Z^* care conţin cuvîntul nul e.

Vom pune în evidenţă o proprietate a limbajelor de tip 2 care ne va permite să arătăm coincidenţa acestei clase cu clasa limbajelor reprezentate în automate AL.

Definiţia II.9.3. O *sp-algebră* este o familie $\mathcal{U}$ de limbaje asupra unui alfabet finit I închisă la operaţiile de reuniune şi produs, adică $L_1, L_2 \in \mathcal{U}$ implică $L_1 \cup L_2, L_1 L_2 \in \mathcal{U}$; *sp-algebra* $\mathcal{U}(L)$ *generată de familia de limbaje* $L = \{L_j \mid j \in J\}$ este clasa limbajelor care se pot obţine prin aplicarea de un număr finit de ori a operaţiilor de reuniune şi produs asupra limbajelor $L = \{L_j \mid j \in J\}$.

Dacă $\mid J \mid < \infty$ sp-algebra $\mathcal{U}(L)$ se zice *finit generată.*

Definiţia II.9.4. O *sp-algebră* $\mathcal{U}$ este *secvenţială* dacă $\forall L \in \mathcal{U}$ şi $i \in I$, $\partial_i(L) \in \mathcal{U}$, unde

$$\partial_i(L) = \{r \mid ir \in L\}.$$

Aplicaţia $\partial_i \mathcal{P}(I^*) \to \mathcal{P}(I^*)$ se numeşte *derivare la stînga în raport cu i* şi se observă că:

$$\partial_i(L_1 \cup L_2) = \partial_i(L_1) \cup \partial_i(L_2), \tag{II.9.1}$$

şi

$$L = \bigcup_{i \in I} i\partial_i L \cup \varepsilon(L), \tag{II.9.2}$$

unde

$$\varepsilon(L) = \begin{cases} \emptyset, & \text{dacă } e \notin L \\ e, & \text{dacă } e \in L, \end{cases}$$

iar $\partial_i(L_1 L_2) = \partial_i(L_1) L_2 \cup \varepsilon(L_1) \partial_i(L_2)$, ceea ce în cazul $e \notin L_1$ revine la

$$\partial_i(L_1 L_2) = \partial_i(L_1) L_2. \tag{II.9.3}$$

Observaţia II.9.3. Generatorii unei sp-algebre secvenţiale, finit generate, sînt în mod necesar limbaje de tip 2. Într-adevăr, fie $\{L_1, \ldots, L_n\}$ un sistem de generatori ai algebrei $\mathcal{U}$. Avem $\partial_i L_k \in \mathcal{U}$ şi deci $\partial_i L_k = \varphi_{ik}(L_1, \ldots, L_n)$, oricare ar fi $i \in I$, $k = \overline{1, n}$ şi cum $L_k = \bigcup_{i \in I} i\partial_i(L_k) \cup \varepsilon(L_k)$ obţinem

$$L_k = \bigcup_{i \in I} i\varphi_{ik}(L_1, \ldots, L_n) \cup \varepsilon(L_k), \quad k = \overline{1, n},$$

unde φ_{ik} sînt funcţii standard.

Ecuaţiile de mai sus constituie un sistem de forma (II.7.2) şi cum sistemul are soluţie unică (în baza observaţiei I.5.6), ţinînd seama de teorema II.7.3, componentele acestei soluţii sînt limbaje de tip 2.

Observaţia II.9.4. Dacă sp-algebra $\mathcal{U}$ este secvenţială atunci orice limbaj $L_0 \in \mathcal{U}$ este reprezentat în semiautomatul $A = (I, \mathcal{U}, f, L_0)$ prin mulţimea de stări $\{L \mid L \in \mathcal{U}, e \in L\}$, iar $f(L, i) = \partial_i(L)$, $\forall L \in \mathcal{U}$, $\forall i \in I$.

Într-adevăr, dacă $p \in L_0$ atunci $e \in f(L_0, p)$ deoarece $\partial_p(p) = e$, iar dacă $e \in f(L_0, p)$ avem $p \in L_0$.

Vom studia sistemul de ecuaţii:

(II.9.4) $$E_j = F_j(E_1, ..., E_n, L_1, ..., L_n),\ j = \overline{1, n},$$

unde $F_j(E_1, ..., E_n, L_1, ..., L_n)$, $j = \overline{1, n}$ sînt funcţii standard peste un alfabet I (v. definiţia II.7.2).

Soluţia minimală a sistemului (II.9.4) este, conform teoremelor II.7.2 şi II.7.6

$$D_j = \bigcup_{h=0}^{\infty} D_j^{(h)},$$

unde $D_j^{(h)} = F_j(D_1^{(h-1)} ...., D_n^{(h-1)}, D_1^{(h-1)} ..... D_n^{(h-1)})$, iar

$$D_j^{(0)} = F_j(\emptyset ..... \emptyset, \emptyset ..... \emptyset).$$

Să observăm că această soluţie se mai poate scrie :

(II.9.5) $$D_j = \bigcup_{k=1}^{\infty} F_j^{(k)}(\emptyset, ... \emptyset, \emptyset, ..., \emptyset),$$

unde $F_j^{(k)} = F_j(F_1^{(k-1)} ..... F_n^{(k-1)}, F_1^{(k-1)} ...., F_n^{(k-1)})$, iar

$F_j^{(1)} = F_j$.

Fie acum sistemul :

(II.9.6) $$L_j = F_j(\xi_1 ...., \xi_n, L_1 ...., L_n),\ j = \overline{1, n}$$

şi $\{\xi_1 ...., \xi_n\} \cap I = \emptyset$.

Să notăm $F_j^{[m+1]} = F_j(\xi_1 ..., \xi_n, F_1^{[m]} ..., F_n^{[m]})$, unde $F_j^{[1]} = F_j$. Soluţia minimală a sistemului (II.9.6) este dată, în baza aceloraşi teoreme de relaţia :

$$P_j = \bigcup_{h=0}^{\infty} P^{(h)}$$

unde $P_j^{(h)} = F_j(\xi_1 ..... \xi_n, P_1^{(h-1)}, ..., P_n^{(h-1)})$ şi

$$P_j^{(0)} = F_j(\xi_1 ...., \xi_n, \emptyset ..... \emptyset).$$

Această soluţie se mai poate scrie :

$$P_j = \bigcup_{k=1}^{\infty} F_j^{[k]}(\xi_1 ...., \xi_n, \emptyset, ..., \emptyset)$$

cu $F_j^{[k+1]} = F_j(\xi_1 ...., \xi_n, F_1^{[k]} ...., F_n^{[k]})$, iar $F_j^{[1]} = F_j$.

Să notăm cu $\overline{Q}$ limbajul care provine din limbajul $Q \subseteq (\{\xi_1 ..... \xi_n\} \cup I)^*$ prin substituirea simbolilor ξ_j cu D_j, $j = \overline{1, n}$.

Au loc :

Teorema II.9.1. *Avem* :

$$\overline{F_j^{[m]}(\xi_1 ...., \xi_n, \emptyset, ..., \emptyset)} = F_j^{[m]}(D_1 ..... D_n, \emptyset, ..., \emptyset).$$

Demonstrație. Vom proceda prin inducție după m.
Pentru $m = 1$ obținem

$$\overline{F_j(\xi_1, ..., \xi_n, \emptyset, ..., \emptyset)} = F_j(D_1, ..., D_n, \emptyset, ..., \emptyset).$$

relație adevărată deoarece exprimă convenția de notație făcută mai sus.
Să presupunem acum relație adevărată pentru m. Atunci avem

$$\overline{F_j^{[m+1]}(\xi_1, ..., \xi_n, \emptyset ..... \emptyset)} = \overline{F_j(\xi_1, ..., \xi_n, F_1^{[m]}(\xi_1, ..., \xi_n, \emptyset, ..., \emptyset)}$$

$$..., \overline{F_n^{[m]}(\xi_1, ..., \xi_n, \emptyset ..... \emptyset)}) = F_j(D_1 ..... D_n, F_1^{[m]}(D_1 ..... D_n, \emptyset, ..., \emptyset), ...$$

$$... F_n^{[m]}(D_1 ..... D_n, \emptyset, ... \emptyset)) = F_j^{[m+1]}(D_1 ..... D_n, \emptyset, ..., \emptyset). \blacksquare$$

Teorema II.9.2. *Sînt valabile incluziunile :*

$$F_j^{(k)}(\emptyset, ... \emptyset, \emptyset, ... \emptyset) \subseteq F_j^{(k)}(D_1 ..... D_n, \emptyset, ..., \emptyset) \subseteq \tag{II.9.7}$$

$$\subseteq F_j^{(k)}(D_1 ..... D_n, D_1 ..... D_n). \quad \forall k \in N^*, \ j = \overline{1, n}.$$

Demonstrație. Să observăm mai întîi că :

$$F_j^{(k)}(D_1, ..., D_n, D_1, ..., D_n) = F_1(D_1, ..., D_n, D_1, ..., D_n) = D_j, \quad \forall j = \overline{1, n}$$

$\forall k \in N^*$, deoarece $(D_1, ..., D_n)$ este soluția sistemului (II.9.4).
Vom proceda în continuare prin inducție după k.
Pentru $k = 1$ partea dreaptă a relației (II.9.7) revine la :

$$F_j(D_1 ..... D_n, \emptyset, ... \emptyset) \subseteq F_j(D_1, ..., D_n, D_1 ..... D_n)$$

adevărată în baza proprietății de monotonie a funcțiilor standard (v. teorema II.9.1).
Presupunem relația (II.9.7) adevărată pentru k. Avem :

$$F_j^{[k+1]}(D_1 ..... D_n, \emptyset, ..., \emptyset) =$$

$$= F_j(D_1 ..... D_n, F_1^{[k]}(D_1 ..... D_n, \emptyset ..... \emptyset), ..., F_n^{[k]}(D_1, ..., D_n, \emptyset, ..., \emptyset)) \subseteq$$

$$\subseteq F_j(D_1 ..... D_n, F_1^{(k)}(D_1, ..., D_n, D_1, ... D_n), ... F_n^{(k)}(D_1 ..... D_n,$$

$$D_1, ... D_n)) = F_j(F_1^{(k)}(D_1 ..... D_n, D_1, ..., D_n), ... F_n^{(k)}(D_1, ... D_n, D_1 ....$$

$$.... D_n), F_1^{(k)}(D_1 ..... D_n, D_1, ..., D_n), ..., F_n^{(k)}(D_1, ..., D_n, D_1 ..... D_n)) =$$

$$= F_j^{(k+1)}(D_1 ..... D_n, D_1, ..., D_n).$$

Procedăm prin inducție după k și pentru a demonstra partea din stînga a relației (II.9.7).
Pentru $k = 1$ verificarea este banală. Presupunem incluziunea adevărată pentru k. Avem :

$F_j^{(k+1)}(\varnothing,...,\varnothing,\varnothing,...,\varnothing) = F_j(F_1^{(k)}(\varnothing,...,\varnothing,\varnothing,...,\varnothing),...,F_n^{(k)}(\varnothing,...,\varnothing,\varnothing,...,\varnothing),F_1^{(k)}(\varnothing,...,\varnothing,\varnothing,...,\varnothing),...,F_n^{(k)}(\varnothing,...,\varnothing,\varnothing,...,\varnothing)) \subseteq F_j(D_1,...,D_n,F_1^{[k]}(D_1,...,D_n,\varnothing,...,\varnothing),...,F_n^{[k]}(D_1,...,D_n,\varnothing,...,\varnothing)) = F_j^{[k+1]}(D_1,...,D_n,\varnothing,...,\varnothing)$.

Aici am folosit faptul că $F_j^{(k)}(\varnothing,...,\varnothing,\varnothing,...,\varnothing) \subseteq D_j$, $\forall k \in N^*$, $j = \overline{1,n}$. ■

Teorema II.9.3. *Are loc relația* $\overline{P}_j = D_j$, $j = \overline{1,n}$.

Demonstrație. Ținînd seama de teoremele precedente și de monotonia *funcțiilor standard avem*:

$$\overline{P}_j = \overline{\bigcup_{k=1}^{\infty} F_j^{[k]}(\xi_1,....\xi_n,\varnothing,...,\varnothing)} = \bigcup_{k=1}^{n} F_j^{[k]}(D_1,...,D_n,\varnothing,...,\varnothing) \subseteq$$

$$\subseteq \bigcup_{k=1}^{\infty} F_j^{(k)}(D_1,....D_n,D_1.....D_n) = F_j(D_1,....D_n,D_1,...,D_n) = D_j, \forall j = \overline{1,n}.$$

Invers, $D_j = \bigcup_{k=1}^{\infty} F_j^{(k)}(\varnothing,...,\varnothing,\varnothing,...,\varnothing) \subseteq \bigcup_{k=1}^{\infty} F_j^{[k]}(D_1,...,D_n,\varnothing,...\varnothing) =$

$$= \overline{\bigcup_{k=1}^{\infty} F_j^{[k]}(\xi_1,....,\xi_n,\varnothing,...,\varnothing)} = \overline{P}_j.$$ ■

Teorema II.9.4. *Limbajul D este de tip 2 dacă și numai dacă el aparține unei sp-algebre secvențiale cu număr finit de generatori.*

Demonstrație. Să presupunem mai întîi că limbajul D este de tip 2, generat de gramatica $G = (I_N, I_T, x_0; F)$ și $e \notin D$ (dacă $e \in D$ și $\mathcal{U}$ ar fi sp-algebra secvențială cu $D \setminus \{e\} \in \mathcal{U}$ atunci algebra $\mathcal{U}'$ construită prin adăugarea la sistemul de generatori ai lui $\mathcal{U}$ a limbajelor $\{e\}$ și $\varnothing$ rămîne o sp-algebră secvențială și $D \in \mathcal{U}'$).

Conform observației II.7.1 limbajul D este prima componentă a soluției minimale a sistemului:

$$x_j = \bigcup_{i_k \in I_T}^{\infty} i_k F_{jk}(x_1,.... x_n) \cup \bigcup_{x_h \in I_N} x_h G_{jk}(x_1,.... x_n),\ j = \overline{1,n}. \tag{II.9.8}$$

Fie deci $D = D_1$, unde $(D_1,...., D_n)$ este soluția minimală a sistemului de mai sus.

Să considerăm alfabetul $\{\xi_1,.... \xi_n\}$ disjunct de I și sistemul de ecuații:

$$Y_j = \bigcup_{i_k \in I_T} i_k F_{jk}(\xi_1,...., \xi_n) \cup \bigcup_{k=1}^{n} Y_k G_{jk}(\xi_1,...., \xi_n),\ j = \overline{1,n},$$

care este liniar în $Y_1,..., Y_n$ pe alfabetul $I_T \cup \{\xi_1,..., \xi_n\}$.

Observînd că limbajele $F_{jk}(\xi_1,..., \xi_n)$, $G_{jk}(\xi_1,..., \xi_n)$ sînt finite urmează că ele sînt limbaje regulate și atunci soluția minimală a sistemului are drept componente limbaje regulate $(P_1, P_2,..., P_n)$ (v. teorema II.4.18).

Fie $A_h = (I_T \cup \{\xi_1, \ldots, \xi_n\}, S_h, f_h; s_{0h})$ semiautomatul inițial în care limbajul regulat este reprezentat prin mulțimea de stări finale W_h, $h = \overline{1, n}$. În automatul $A = \prod_{h=1}^{n} A_h = (I_T \cup \{\xi_1, \ldots, \xi_n\}, \prod_{h=1}^{n} S_h, f; s_0 = (s_{01}, s_{02}, \ldots, s_{0n}))$, unde $f((s_1, \ldots, s_n), y) = (f_1(s_1, y), \ldots, f_n(s_n, y))$, $\forall y \in I_T \cup \{\xi_1, \ldots, \xi_n\}$, $(s_1, \ldots, s_n) \in \prod_{h=1}^{n} S_h$, limbajele P_h, $h = \overline{1, n}$ sînt reprezentate respectiv de mulțimile de stări $V_h = \{s = (s_1, \ldots, s_n), s_h \in W_h\}$.

Să considerăm limbajul $Q_{s,v_h} = \{p \mid p \neq e, f(s, p) \in V_h\}$.

Este clar că $P_h = Q_{s_0,v_h}$. Vom nota cu $\bar{Q}$ limbajul obținut din limbajul $Q \subseteq (I_T \cup \{\xi_1, \ldots, \xi_n\})^*$ prin substituirea simbolilor $\xi_1, \ldots, \xi_n$ prin limbajele $D_1, \ldots, D_n$ respectiv.

În baza teoremei II.9.3 avem $D_h = \bar{P}_h = \bar{Q}_{s_0,v_h}$ $h = \overline{1, n}$.

Vom arăta că sp-algebra $\mathcal{U}$ generată de limbajele Q_{s_0,v_h} este secvențială. Avem:

$$Q_{s,v_h} = \bigcup_{i_k \in I_T} i_k [Q_{f(s,i_k),v_h} \cup \varepsilon(f(s, i_k), V_h)] \cup$$

$$\cup \bigcup_{k=1}^{n} \xi_k Q_{f(s,\xi_k),v_h} \cup \varepsilon(f(s, \xi_k), V_h),$$

unde

$$\varepsilon(s, V_h) = \begin{cases} e, \text{ dacă } s \in V_h \\ \varnothing, \text{ în caz contrar.} \end{cases}$$

Din relațiile (II.9.8) și formulele (II.9.5) urmează că toate cuvintele limbajului P_h încep cu litere din alfabetul I_T și prin urmare $Q_{f(s_0,\xi_k),v_h} = \varnothing$.

Rămîne deci $Q_{s_0,v_h} = \bigcup_{i_k \in I_T} i_k [Q_{f(s_0,i_k),v_h} \cup \varepsilon(f(s_0, i_k), V_h)]$.

Apoi:

$$\bar{Q}_{s,v_h} = \bigcup_{i_k \in I_T} i_k [\bar{Q}_{f(s,i_k),v_h} \cup \varepsilon(f(s, i_k), V_h)] \cup$$

$$\cup \bigcup_{k=1}^{n} \bar{\xi}_k [\bar{Q}_{f(s,\xi_k),v_h} \cup \varepsilon(f(s, \xi_k), V_h)].$$

Deoarece $\bar{\xi}_k = D_k = \bar{P}_h = \bar{Q}_{s_0,v_h}$ avem

$$Q_{s,v_h} = \bigcup_{i_k \in I_T} i_k \bar{Q}_{f(s,i_k),v_h} \cup \varepsilon(f(s, i_k), V_h)] \cup$$

$$\bigcup_{k=1}^{n} [\bigcup_{i_j \in I_T} i_j (\bar{Q}_{f(s_0,i_j),v_k} \cup \varepsilon(f(s_0, i_j), V_k))] (\bar{Q}_{f(s,\xi_k),v_h} \cup \varepsilon(f(s, i_k), V_h)).$$

În sfîrșit obținem :

$$\partial_{i_l} \bar{Q}_{s,V_h} = \bar{Q}_{f(s,\,i_l)\;\;V_h} \cup \varepsilon(f(s, i_l), V_h) \cup \bigcup_{k=1}^{n} [\bar{Q}_{f(s_0,\,i_l),\,V_k} \cup \tag{II.9.9}$$

$$\cup\, \varepsilon(f(s_0, i_l), V_k)]\,[\bar{Q}_{f(s,\,\xi_k),\,V_h} \cup \varepsilon(f(s, i_k), V_h)],$$

și ținînd seama de relațiile (II.9.1 – 3) derivata unui limbaj E din $\mathcal{U}$ aparține algebrei $\mathcal{U}$ ceea ce arată că sp-algebra $\mathcal{U}$ este secvențială.

Să presupunem acum că $\mathcal{U}$ este o sp-algebră secvențială generată de limbajele $E_1, E_2, ..., E_n$ unde $E_k \subseteq I^*$, $k = \overline{1, n}$. Vom arăta că oricare limbaj $E \in \mathcal{U}$ este de tip 2. Să observăm mai întîi că $E = \varphi(E_1, ..., E_n)$ unde φ este o funcție standard, $\varphi : (\mathcal{P}(I^*))^n \to \mathcal{P}(I^*)$.

Putem scrie :

$$E_h = \bigcup_{i \in I} i\,(\partial_i E_h \cup \varepsilon(E_h)),$$

și cum $\mathcal{U}$ este o sp-algebră secvențială $\partial_i E_h \in \mathcal{U}$, deci $\partial_i E_h = \varphi_{ih}(E_1, ..., E_n)$, unde φ_{ih} sînt funcții standard.

Sistemul de ecuații

$$x_0 = \varphi(x_1, ..., x_n)$$

$$x_h = \bigcup_{i \in I} i_i(\varphi_{ih}(x_1, ..., x_n) \cup \varepsilon(D_h)], \quad h = \overline{1, n}.$$

admite soluția $(E, E_1, ..., E_n)$.

Mai mult, soluția sistemului (II.9.10) este unică, ceea ce rezultă din teorema I.5.9 și din observația I.5.6 și deci ea este și soluția minimală. De aceea prima ei componentă – limbajul E – este un limbaj de tip 2. ■

Teorema II.9.5. *Un limbaj $E \subseteq I^*$ este de tip 2 dacă și numai dacă limbajul E este reprezentat într-un AL* (v. definiția II.9.2).

Demonstrație. Fie E un limbaj de tip 2 și să presupunem mai întîi că $e \notin E$. Să notăm prin $\mathcal{U}$, sp-algebra secvențială construită în demonstrația teoremei precedente cu $E \in \mathcal{U}$.

Mulțimea finită de limbaje $\{\bar{Q}_s, V_h\}$ care generează sp-algebra $\mathcal{U}$ este pusă în corespondență biunivocă cu alfabetul finit Z prin bijecția : $\{\bar{Q}_{s,V_h}\} \to Z$, unde

$$\alpha(\bar{Q}_s, V_h) = z_{s,V_h}.$$

Considerăm automatul de tip Leticevski $AL = (I, \mathcal{P}_f(Z^*), F ; z_{s_0}, V_1)$ unde definim $F(\{z_{s,V_h}\}, i) = \alpha^{-1}(\partial_i \bar{Q}_{s,V_h})$. Să observăm că funcția F satisface condițiile din definiția II.9.1. Ținînd seama de expresia (II.9.9) avem :

(I I.9.10) $$F(\{z_{s,V_h}\}, i) = \{z_{f(s,i),V_h} \cup \varepsilon(f(s,i), V_h)\} \cup$$

$$\cup \bigcup_{k=1}^{n} [\{z_{f(s_0,i),V_h}\} \cup \varepsilon(f_{(s_0,i)V_h})] [\{z_{f(s,\xi_k),V_h}\} \cup \varepsilon(f(s,\xi_k)V_h)].$$

Să observăm că limbajul E este reprezentabil în semiautomatul $A_{\mathcal{U}} = (I, \mathcal{U}, F_{\mathcal{U}}; E)$ unde $F_{\mathcal{U}}(E_1, i) = \partial_i E_1$, $\forall E_1 \in \mathcal{U}$, $i \in I$, prin mulțimea de stări finale $\{E', \mid e \in E'\}$.

Bijecția $\alpha : \{Q_{s,V_h} \mid s \in \prod_{h=1}^{n} S_h,\ V_h \subseteq \prod_{h=1}^{n} S_h\} \to Z$

poate fi extinsă în mod natural la un homomorfism între sp-algebrele $(\mathcal{U}, .., \cup)$ și $(\mathcal{P}(Z^*), ., \cup)$, care va fi notat de asemenea cu α.

Vom arăta că homomorfismul α induce un homomorfism surjectiv de semiautomate $\alpha : A_{\mathcal{U}} \to AL$, deoarece $\alpha(F_{\mathcal{U}}(E_1, i)) = F(\alpha(E_1), i)$, $\forall E_1 \in \mathcal{U}$, $i \in I$. Pentru aceasta va fi suficient să arătăm că $\alpha(F_{\mathcal{U}}(\bar{Q}_{s,V_h}, i)) = F(\alpha(\bar{Q}_{s,V_h}), i)$ sau $\alpha(\partial_i \bar{Q}_{s,V_h}) = F(\{z_{s,V_h}\}, i)$ ceea ce are loc ținînd seama de relațiile (II.9.9) și (II.9.10). Homomorfismul $\alpha : \mathcal{U} \to \mathcal{P}(Z^*)$ duce un limbaj care conține cuvîntul nul e într-un limbaj care conține pe e. Prin urmare limbajul E, de tip 2, va fi reprezentat în automatul AL.

Dacă $e \in E$ atunci în automatul AL în care se reprezintă limbajul $E \setminus \{e\}$ adăugăm stării inițiale cuvîntul nul e și obținem un nou automat AL în care limbajul E este reprezentat.

Invers, să presupunem că limbajul $E = L(AL)$ unde $AL = (I, \mathcal{P}_f(Z^*), F; L_0)$. Vom arăta că limbajul E aparține unei sp-algebre $\mathcal{U}$ secvențiale finit generate, de unde va rezulta, în baza teoremei II.9.4 că limbajul E este de tip 2.

Pentru fiecare $Q \in \mathcal{P}^f(Z^*)$ fie limbajul:

$$E_Q = \{p \mid p \in I^*,\ e \in F(Q, p)\}.$$

$$E_{L_0} = \{p \mid p \in I^*,\ e \in F(L_0, p)\} = E \text{ și}$$

Avem $$E_{Q_1 \cup Q_2} = E_{Q_1} \cup E_{Q_2}.$$

$$E_{Q_1 Q_2} = E_{Q_1} E_{Q_2}.$$

$$E_{F(Q,i)} = \partial_i E_Q.$$

Ca atare sp-algebra secvențială generată de limbajele $\{E_{\{z\}} \mid z \in Z\} \cup \{\{e\}, \varnothing\}$ are un număr finit de generatori și E aparține acestei sp-algebre. ■

§ II. 10. LIMBAJE DE TIP 1 (LIMBAJE SENSIBILE LA CONTEXT)

Vom prezenta în acest paragraf proprietăţile limbajelor de tip 1.

Observaţia II.10.1. Conform definiţiei II.2.1 pentru o deducţie $q_1 \overset{*}{\underset{G}{\Rightarrow}} q_2$ avem $l(q_1) < l(q_2)$, dacă $q_2 \neq e$.

În baza acestei observaţii clasa limbajelor de tip 1 este inclusă în clasa limbajelor generate de gramaticile descrise în :

Definiţia II.10.1. O *gramatică expansivă* [1] este o gramatică $G = (I_N, I_T, x_0; F)$ pentru care orice regulă de generare $(r_1 \to r_2) \in F$ are proprietatea $l(r_2) \geqslant l(r_1)$ sau, dacă $r_2 = e$, atunci $r_1 \in I_N$ şi r_1 nu apare în partea dreaptă a unei reguli de generare din F.

Pentru a stabili legătura dintre gramaticile de tip 1 şi gramaticile expansive, este necesară în prealabil :

Teorema II.10.1. *Fie $G = (I_N, I_T, x_0; F)$ o gramatică de tip 1 care nu conţine reguli de generare de forma $x \to e$, $x \in I_N$. Fie $r_1, r_2 \in (I_N \cup I_T)^*$, astfel încît r_2 să conţină cel puţin un simbol din I_N şi $l(r_1) \leqslant l(r_2)$. Atunci, pentru gramatica $G_1 = (I_N, I_T, x_0; F \cup \{r_1 \to r_2\})$ există o gramatică de tip 1 echivalentă.*

Demonstraţie. Dacă $r_1 = x_1 x_2 \ldots x_m$ şi $r_2 = x_1' x_2' \ldots x_n'$ cu $x_j, x_k' \in I_N$, $j = \overline{1, m}$, $k = \overline{1, n}$, atunci adăugăm alfabetului I_N noi simboli neterminali $z_1 \ldots z_m$, iar lui F regulile de generare :

$$
(R) \quad \left\{ \begin{array}{l}
x_1 x_2 \ldots x_m \to z_1 x_2 \ldots x_m \\
z_1 x_2 \ldots x_m \to z_1 z_2 x_3 \ldots x_m \\
\quad \cdot \\
\quad \cdot \\
\quad \cdot \\
z_1 z_2 \ldots z_{m-1} x_m \to z_1 z_2 \ldots z_{m-1} z_m x_{m+1}' \ldots x_n' \\
z_1 z_2 \ldots z_m x_{m+1}' \ldots x_n' \to x_1' z_2 \ldots z_m x_{m+1}' \ldots x_n' \\
\quad \cdot \\
\quad \cdot \\
\quad \cdot \\
x_1' x_2' \ldots x_{m-1}' z_m x_{m+1}' \ldots x_n' \to x_1' x_2' \ldots x_n'
\end{array} \right.
$$

Deoarece regulile de generare din mulţimea R au aspectul prescris regulilor de generare din gramatici de tip 1, gramatica $G' = (I_N \cup \{z_1, \ldots z_m\}, I_T, x_0; F \cup R)$ este de tip 1. Mai mult, G' este echivalentă cu G_1. Dacă $p \in L(G')$ şi în deducţia $x_0 \overset{*}{\underset{G'}{\Rightarrow}} p$ au fost folosite doar reguli din F atunci evident $p \in L(G_1)$.

[1] denumirea în literatura engleză este „length increasing grammar".

Dacă în deducția $x_0 \overset{*}{\underset{G}{\Rightarrow}} p$ a fost folosită prima regulă de generare din R (și în deducție ea este obligatoriu prima regulă din R folosită, cuvintele care apar în regulile din F neconținînd simboli z), atunci *toate* regulile din R apar în deducție în aceeași ordine în care apar în R, deoarece în caz contrar obținerea unui cuvînt terminal n-ar fi posibilă.

Astfel, în gramatica G_1 avem deducția $x_0 \overset{*}{\underset{G_1}{\Rightarrow}} r_1 \underset{G_1}{\Rightarrow} r_2 \overset{*}{\underset{G_1}{\Rightarrow}} p$ și $p \in L(G_1)$.

Invers, incluziunea $L(G_1) \subseteq L(G')$ se obține imediat.

Dacă r_1 și r_2 conțin și simboli din I_T, adică $r_1 = y_1 \dots y_m$ și $r_2 = y'_1 \dots y'_n$, unde $y_j, y'_k \in I_N \cup I_T$, $j = \overline{1,m}$, $k = \overline{1,n}$, atunci vom adăuga alfabetului I_N noi simboli neterminali $\{z_{j_h} \mid h = \overline{1,m_1}\}$, $m_1 \leqslant m$ și $\{z'_{k_l} \mid l = \overline{1,n_1}\}$, $n_1 \leqslant n$, cîte unul de fiecare din simbolii $\{y_j \mid j = \overline{1,m}\} \cup \{y'_k \mid k = \overline{1,n}\}$ care aparțin lui I_T. Fie F' mulțimea de reguli de generare care se obține din $F \cup \{r_1 \to r_2\}$, în care fiecare y_j și y'_k terminal este înlocuit prin z_j - ul sau z'_k -ul respectiv.

Gramatica $G'_1 = (I_N \cup \{z_{j_h} \mid h = \overline{1,m_1}\} \cup \{z'_{k_l} \mid l = \overline{1,n_1}\}, I_T, x_0; F' \cup \{z_j \to y_j \mid y_j \in I_T\} \cup \{z'_k \to y'_k \mid y'_k \in I_T\}$ este o gramatică echivalentă cu G_1.

Într-adevăr, dacă $p \in L(G_1)$ și avem $x_0 \overset{*}{\underset{G_1}{\Rightarrow}} r_1 \overset{*}{\underset{G_1}{\Rightarrow}} r_2 \overset{*}{\underset{G_1}{\Rightarrow}} p$, atunci avem $x_0 \overset{*}{\underset{G'_1}{\Rightarrow}} r''_1 \overset{*}{\underset{G'_1}{\Rightarrow}} r''_2 \overset{*}{\underset{G'_1}{\Rightarrow}} p''$ deducție care se obține din cea precedentă înlocuind simbolii terminali $\{y_j \mid j = \overline{1,m}\} \cup \{y'_k \mid k = \overline{1,n}\}$ cu simboli corespunzători. Această nouă deducție poate fi completată astfel: $x_0 \overset{*}{\Rightarrow} p'' \overset{*}{\Rightarrow} p$, aplicînd regulile de generare de forma $z_j \to y_j$ sau (și) $z'_k \to y'_k$; deci $p \in L(G'_1)$.

Este clar că, dacă $p \in L(G_1)$ și în deducția $x_0 \overset{*}{\underset{G_1}{\Rightarrow}} p$ nu s-a folosit regula de generare $r_1 \to r_2$, atunci $p \in L(G'_1)$, deci $L(G_1) \subseteq L(G'_1)$.

Invers, fie $p \in L(G'_1)$, adică $x_0 \overset{*}{\underset{G'_1}{\Rightarrow}} p$. Fie r''_1 și r''_2 cuvintele obținute din r_1 și r_2 prin înlocuirea simbolilor terminali cu simboli de forma z_j sau z'_k. Dacă în deducția $x_0 \overset{*}{\underset{G_1}{\Rightarrow}} p$ nu s-a utilizat regula de generare $r''_1 \to r''_2$, atunci există în gramatica G_1 deducția $x_0 \overset{*}{\underset{G_1}{\Rightarrow}} p$, în care nu apare regula de generare $r_1 \to r_2$ și $p \in L(G_1)$.

Dacă $x_0 \overset{*}{\underset{G'_1}{\Rightarrow}} r'_1 \overset{*}{\underset{G'_1}{\Rightarrow}} r''_2 \overset{*}{\underset{G'_1}{\Rightarrow}} p$ avem $x_0 \overset{*}{\underset{G_1}{\Rightarrow}} r_1 \underset{G_1}{\Rightarrow} r_2 \overset{*}{\underset{G_1}{\Rightarrow}} p$ și $p \in L(G_1)$.

Să observăm că gramatica G'_1, echivalentă cu G_1, nu este de tip 1, dar are forma gramaticii G_1 din prima parte a demonstrației.

Există deci o gramatică de tip 1 G'' echivalentă cu G'_1, deci echivalentă cu G_1. ■

Observația II.10.2. Procedînd analog ca în demonstrația teoremei precedente, se poate arăta că pentru gramatica $G_2 = (I_N, I_T, x_0; F \cup \{r_1 \to r_1', \ldots, r_m \to r_m'\})$ cu $l(r_j) \leqslant l(r_j')$ și r_j conține măcar un simbol din I_N, $j = \overline{1, m}$, există o gramatică de tip 1, echivalentă.

Teorema II.10.2. *Pentru orice gramatică expansivă există o gramatică de tip* 1, *echivalentă.*

Demonstrație. Putem presupune că gramaticile care vor apare aici nu conțin reguli de generare de forma $x \to e$, deoarece în caz contrar putem îndepărta aceste reguli și introduce un simbol inițial y_0; însoțit de regula de generare $y_0 \to e$.

Fie $G = (I_N, I_T, x_0; F)$ o gramatică expansivă. Atunci $F = F' \cup \{(r_j \to r_j') \mid l(r_j) \leqslant l(r_j'), j = \overline{1, m}\}$, iar regulile de generare din F' satisfac la, restricțiile impuse unei gramatici de tip 1. Dacă $F' = \varnothing$, atunci și $L(G) = \varnothing$ deoarece nu există nici o regulă de generare de forma $x_0 \to r$. În acest caz există evident o gramatică de tip 1, $G_1 = (I_N, I_T, x_0; \varnothing)$ echivalentă cu G.

Dacă $F' \neq \varnothing$, atunci, ținînd seama de observația II.10.2, există o gramatică de tip 1, echivalentă cu gramatica expansivă G. ■

Un alt tip de gramatici legate de gramaticile de tip 1 este dat în definițiile de mai jos:

Definiția II.10.2. O gramatică cu marcaj final # este o gramatică:

$$G_{\#} = (I_N \cup \{\#\}, I_T, x_0; F).$$

astfel încît $\# \notin I_N \cup I_T$ și fiecare regulă de generare din F este de una din formele:

$$r_1 \to r_2, \quad \#r_1 \to \#r_2, \quad r_1\# \to r_2\#, \tag{II.10.1}$$

unde $r_1, r_2 \in (I_N \cup I_T)^*$ și r_1 conține cel puțin un simbol din I_N.

Definiția II.10.3. *O gramatică de tip 1 cu marcaj final* # este o gramatică cu marcaj final #, în care regulile de generare obținute din (II.10.1) prin îndepărtarea marcajului final #, satisfac restricțiile impuse gramaticilor expansive.

Definiția II.10.4. Limbajul $L(G, \#)$ generat de gramatica $G_{\#}$ este

$$L(G, \#) = \{p \mid p \in I_T^*, \ \# x_0 \# \overset{*}{\underset{G_{\#}}{\Rightarrow}} \# p \#\}.$$

Remarcăm că $L(G, \#) \neq L(G_{\#})$ întrucît în $L(G, \#)$ cuvintele apar fără marcaj final.

Observația II.10.3. Noțiunea de echivalență între gramatici cu marcaj final este asemănătoare celei obișnuite. Astfel, gramaticile $G_{\#}^1$ și $G_{\#}^2$ sînt echivalente dacă $L(G', \#) = L(G^2, \#)$.

Observația II.10.4. Pentru orice gramatică de tip 1 există, în baza teoremei II.10.2, o gramatică de tip 1 cu marcaj final, echivalentă.

Rezultatul reciproc este conținut în:

Teorema II.10.3. *Pentru orice gramatică de tip* 1 *cu marcaj final există o gramatică de tip 1 echivalentă.*

Demonstrație. Fie $G_{\#} = (I_N \cup \{\#\}, I_T, x_0; F)$ o gramatică de tip 1 cu marcaj final $\#$. Putem presupune că nu apar regulile de generare de forma $x \to e$ prin îndepărtarea marcajului final.

Pentru fiecare simbol $y \in I_N \cup I_T = I$ se introduc trei simboli: $\#y$, $y\#$, $\#y\#$ și fie:

$\#I = \{\#y \mid y \in I\}$, $I\# = \{y\# \mid y \in I\}$, $\#I\# = \{\#y\# \mid y \in I\}$, iar pentru

$r = y_1 y_2 \dots y_{n-1} y_n$, $n \geqslant 1$, $y_j \in I$, $j = \overline{1, n}$ fie

$\#r = (\#y_1) y_2 \dots y_{n-1} y_n$, $r\# = y_1 y_2 \dots y_{n-1} (y_n\#)$,

$$r = \begin{cases} (\#y_1) y_2 \dots y_{n-1} (y_n\#) \text{ pentru } n > 1 \\ \#y_1\# \text{ pentru } n = 1. \end{cases}$$

Vom arăta că gramatica expansivă $G_1 = (I_N \cup \#I \cup I\# \cup \#I\#, I_T, \#x_0\#; F_1)$, unde F_1 constă din regulile de generare obținute prin următoarele procedee:

1) dacă $(r_1 \to r_2) \in F$ atunci $r_1 \to r_2$, $\#r_1 \to \#r_2$, $r_1\# \to r_2\#$ și $\#r_1\# \to \#r_2\#$ aparțin lui F_1;

2) dacă $(\#r_1 \to \#r_2) \in F$ atunci $\#r_1 \to \#r_2$ și $\#r_1\# \to \#r_2\#$ sînt în F_1;

3) dacă $(r_1\# \to r_2\#) \in F$ atunci $r_1\# \to r_2\#$ și $\#r_1\# \to \#r_2\#$ sînt în F_1;

4) $\#y \to y$, $y\# \to y$ și $\#y\# \to y$ sînt în F_1, pentru orice $y \in I$,

unde $r_1, r_2 \in I^* \setminus \{e\}$, este echivalentă cu gramatica $G_{\#}$, adică $L(G_1) = L(G, \#)$.

Într-adevăr, dacă $\#x_0\# \overset{\cdot}{\underset{G_1}{\Rightarrow}} p$ există o deducție

$\#x_0\# \underset{G_{\#}}{\Rightarrow} \#p\#$, deci $L(G_1) \subseteq L(G, \#)$ și invers, dacă $\#x_0\# \overset{\cdot}{\underset{G_{\#}}{\Rightarrow}} \#p\#$ există

o deducție $\#x_0\# \overset{\cdot}{\underset{G_1}{\Rightarrow}} p$ și $L(G, \#) \subseteq L(G_1)$.

Conform teoremei II.10.2, există o gramatică G', de tip 1 echivalentă cu gramatica expansivă G_1 și deci echivalentă cu gramatica $G_{\#}$. ■

Teorema II.10.4. *Fie G o gramatică expansivă (de tip 1). Există o gramatică expansivă (de tip 1) G' echivalentă cu G, astfel încît regulile de generare ale lui G', care conțin simboli terminali, sînt de forma $x \to i$, $x \in I_N$, $i \in I_T$.*

Demonstrație. Vom înlocui în regulile de generare ale gramaticii $G = (I_N, I_T, x_0; F)$ fiecare simbol terminal i printr-un nou simbol neterminal x_i. Gramatica G' va fi $G' = (I_N \cup \{x_i \mid i \in I_T\}, I_T, x_0; F')$, unde F' se obține din F prin procedeul indicat mai sus și prin adăugarea regulilor de generare $(x_i \to i)$; ea este echivalentă cu G și de același tip cu G (exercițiu!). ■

Definiția II.10.5. O gramatică de ordinul n, $n \geqslant 1$ este o gramatică $G = (I_N, I_T, x_0; F)$, pentru care $\forall (r_1 \to r_2) \in F$, $l(r_1) \leqslant n$ și $l(r_2) \leqslant n$.

Teorema II.10.5. *Pentru orice gramatică expansivă de ordinul $n \geqslant 3$ există o gramatică expansivă echivalentă de ordin $n - 1$.*

Demonstrație. Fie $G_n = (I_N, I, x_0; F)$ o gramatică expansivă de ordin $n \geqslant 3$, în care putem presupune, fără a restrînge generalitatea, că regulile de generare care conțin simboli terminali sînt de forma $x \to i$ (v. teorema II.10.4). Vom construi acum gramatica expansivă G_{n-1} de ordin $n - 1$ echivalentă cu G_n. Fie deci $G_{n-1} = (I_N, I_T, x_0; F')$, unde F' se construiește astfel:

1) dacă $(r_1 \to r_2) \in F$ și $l(r_2) \leqslant 2$, atunci avem și $l(r_1) \leqslant 2$ și luăm $(r_1 \to r_2) \in F'$.

2) dacă $(r_1 \to r_2) \in F$ și $l(r_2) \geqslant 3$, atunci există $x_1, x_2, x_3, x_4 \in I_N$, astfel ca $r_1 = x_1 r_1'$, $r_2 = x_2 x_3 x_4 r_2'$ cu $r_1', r_2' \in I_N^*$ (eventual putînd fi cuvinte nule).

În această situație distingem două cazuri:

2a) dacă $r_1' = e$, atunci $r_1 = x_1$ și introducem doi noi simboli neterminali $z_1 z_2$ și regulile de generare:

$$x_1 \to z_1 z_2, \quad z_1 \to x_2, \quad z_2 \to x_3 x_4 r_2' \in F';$$

2b) dacă $r_1' \neq e$, atunci $r_1' = x_5 r_1''$ cu $r_1'' \in I_N^*$ și $r_1 = x_1 x_5 r_1''$; introducem atunci doi noi simboli neterminali z_3, z_4 și regulile de generare:

$$x_1 x_5 \to z_3 z_4, \quad z_3 \to x_2, \quad z_4 r_1'' \to x_3 x_4 r_2' \in F'.$$

3) regulile de generare din F' se obțin prin aplicarea procedeelor de la punctele 1) și 2), după caz, fiecărei reguli de generare din F.

4) la regulile obținute mai sus se adaugă neschimbate regulile de generare din F de forma $x \to i$.

Este ușor de verificat că $L(G_n) = L(G_{n-1})$. ■

Consecința II.10.1. Pentru orice gramatică expansivă există o gramatică expansivă echivalentă de ordin 2.

Fie G o gramatică expansivă și $n = \max\{l(r_2) \mid (r_1 \to r_2) \in F\}$.

Evident, G este o gramatică de ordin n și prin aplicarea repetată (de $n - 2$ ori) a teoremei precedente, obținem în final o gramatică expansivă echivalentă de ordin 2.

Definiția II.10.6. O *gramatică liniar-mărginită* este o gramatică $G = (I_N, I_T, x_0; F)$, pentru care, dacă $(r_1 \to r_2) \in F$, atunci avem una din următoarele situații:

$$\text{(II.10.2)} \qquad \begin{aligned} &1)\ r_1 = x_0;\ r_2 = x_0 x,\ x \in I_N, \\ &2)\ r_1 = x_1 x_2,\ r_2 = z_1 z_2,\ x_1, x_2, z_1, z_2 \in I_N \\ &3)\ r_1 = x_1,\ r_2 = y,\ y \in I_N \cup I_T,\ x_1 \in I_N \end{aligned}$$

și $x_0 \notin \{y, z_1, z_2, x\}$.

Teorema II.10.6. *Pentru orice gramatică expansivă G există o gramatică liniar-mărginită G', astfel încît $L(G') = L(G)$ sau $L(G') = L(G) \setminus \{e\}$.*

Demonstrație. Putem presupune (în baza teoremei II.10.4 și a consecinței II.10.1), că pentru gramatica $G = (I_N, I_T, x_0; F)$, F conține numai reguli de generare de următoarele forme:

1) $x_0 \to e$,
2) $x \to y$, $x \in I_N$, $y \in I_N \cup I_T$,
3) $x \to z_1 z_2$, unde $x_1, z_1, z_2 \in I_N$,
4) $x_1 x_2 \to z_1 z_2$, unde $x_1, x_2, z_1, z_2 \in I_N$.

Să construim o gramatică liniar-mărginită: $G' = (I_N \cup \{x_0', x_1'\}, I_T, x_0'; F')$, $x_0', x_1' \notin I_N$. În acest scop F' va conține toate regulile de generare ale lui F de la punctele 2) și 4), care satisfac cerințele impuse regulilor de generare din gramaticile liniar-mărginite. Apoi, pentru fiecare regulă de la punctul 3) vom introduce regula de generare $xx_1' \to z_1 z_2$, iar simbolul x_1' mai apare în următoarele reguli:

$x_0' \to x_0' x_1'$, $x_1' x \to x x_1'$, $x x_1' \to x_1' x$, $\forall x \in I_N \cup I_T$. Mai avem și $x_0' \to x_0$.

Întrucît toate regulile de generare au aspectul prescris regulilor de generare (II.10.2), gramatica G' este liniar-mărginită.

Să considerăm acum gramatica G_0, obținută din gramatica G prin eliminarea regulii de generare $x_0 \to e$ (dacă această regulă era în G). Avem:

$$L(G_0) = \begin{cases} L(G), \text{ dacă } (x_0 \to e) \notin F, \text{ adică } e \notin L(G) \\ L(G) \setminus \{e\}, \text{ în caz contrar.} \end{cases}$$

Vom arăta că $L(G_0) = L(G')$, de unde va rezulta valabilitatea teoremei.

Fie $p \in L(G_0)$, adică există deducția $x_0 \overset{\bullet}{\underset{G_0}{\Rightarrow}} p$, în care să presupunem că s-au folosit un număr de n reguli de generare de forma indicată la punctul 3), $n \geqslant 0$.

Ținînd seama de definiția gramaticii G' din deducția precedentă, putem obține o deducție în G' de forma $x_0' \underset{G'}{\Rightarrow} x_0' x_1' \overset{\bullet}{\underset{G'}{\Rightarrow}} x_0' \underbrace{x_1' \dots x_1'}_{n} \underset{G'}{\Rightarrow} x_0 \underbrace{x_1' \dots x_1'}_{n} \overset{\bullet}{\underset{G'}{\Rightarrow}} p$.

Cuvîntul $x_0 \underbrace{x_1' \dots x_1'}_{n}$ avînd n simboli x_1' servește ca un „rezervor" de simboluri x_1', permițînd utilizarea regulilor din G' care înlocuiesc regulile de forma 3) din G_0.

Aici utilizăm, cînd este necesar, comutabilitatea simbolului x_1' cu orice $x \in I_N \cup I_T$ pentru aducerea simbolilor x_1' pe pozițiile necesare.

Avem deci $p \in L(G')$, de unde $L(G_0) \subseteq L(G')$.

Invers, fie $p \in L(G')$, adică există deducția $x_0' \overset{\bullet}{\underset{G'}{\Rightarrow}} p$.

Obținem imediat o deducție în G_0, $x_0 \overset{\bullet}{\underset{G_0}{\Rightarrow}} p$, eliminînd toți simbolii x_1' care apar în deducția din G'. Astfel $L(G') \subseteq L(G_0)$. ■

Consecința II.10.2. Pentru orice gramatică G de tip 1 există o gramatică liniar-mărginită G', astfel încît :

$$L(G') = \begin{cases} L(G), \text{ dacă } e \notin L(G) \\ L(G) \setminus \{e\}, \text{ în caz contrar.} \end{cases}$$

Afirmația rezultă imediat din teorema precedentă și din faptul că orice limbaj de tip 1 poate fi generat de o gramatică expansivă. ■

Observația II.10.5. Fie $G = (I_N, I_T, x_0; F)$ o gramatică liniar-mărginită și $p = i_1 i_2 \ldots i_n \in L(G)$. Există o deducție $x_0 \overset{\bullet}{\underset{G}{\Rightarrow}} p$, care poate fi descrisă cu ajutorul unui hiperarbore, prin analogie cu descrierea prin arbori de deducție (v. definiția II.5.1) a cuvintelor generate într-o gramatică de tip 2.

Definiția II.10.7. *Un hiperarbore de deducție* pentru o gramatică liniar-mărginită este un hiperarbore care satisface următoarele condiții :

1) orice nod este etichetat cu un simbol din $I = I_N \cup I_T$

2) rădăcina hiperarborelui este etichetată cu x_0

3) nodurile care au un descendent propriu sînt etichetate cu simboli din I_N.

4) dacă avem în hiperarbore o muchie de forma din figura II.10.1.a, atunci în deducție se folosește regula de generare $x_0 \to x_0\, x_1$; dacă în hiperarbore apare o muchie de forma descrisă în figura II.10.1.b, atunci, corespunzător, avem în deducție regula $x_1\, x_2 \to z_1\, z_2$, iar pentru o muchie de forma din figura II.10.1.c se utilizează regula de generare $x \to y$.

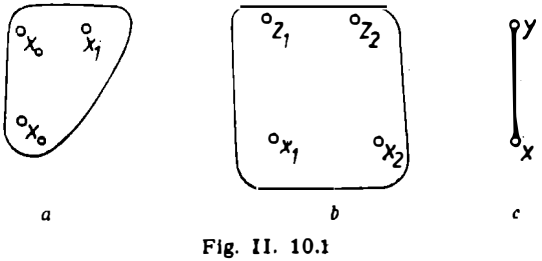

Fig. II. 10.1

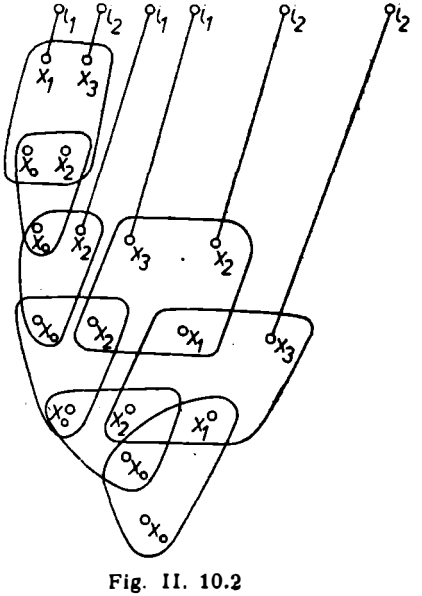

Fig. II. 10.2

5) nodurile terminale ale arborelui în ordinea în care apar descriu cuvîntul generat prin deducția descrisă de hiperarbore.

Exemplul II.10.1. Fie gramatica $G = (\{x_0, x_1, x_2, x_3\}, \{i_1, i_2\}, x_0; F)$, unde $F = \{x_0 \to x_0 x_1,\ x_0 \to x_0 x_2,\ x_0 x_2 \to x_1 x_3,\ x_0 \to i_1,\ x_1 \to i_1,\ x_2 \to i_1,\ x_2 \to i_2,\ x_2 x_1 \to x_1 x_3,\ x_2 x_1 \to x_3 x_2,\ x_3 \to i_1,\ x_3 \to i_2\}$. Hiperarborele din figura II.10.2 corespunde deducției $x_0 \Rightarrow x_0 x_1 \Rightarrow x_0 x_2 x_1 \Rightarrow x_0 x_1 x_3 \Rightarrow x_0 x_2 x_1 x_3 \Rightarrow x_0 x_2 x_2 x_1 x_3 \Rightarrow x_0 x_2 x_3 x_2 x_3 \Rightarrow x_0 x_2 x_2 x_3 x_2 x_3 \Rightarrow x_1 x_3 x_2 x_3 x_2 x_3 \Rightarrow i_1 i_2 i_1 i_1 i_2 i_2$.

Remarcăm că lungimea unui cuvînt $p \in L(G)$ este determinată de numărul m de aplicări ale regulilor de generare de forma $x_0 \to x_0 x$ și fiind anume $m + 1$.

Dacă un nod este etichetat cu simbolul x_0, există un lanț care unește acest nod cu vîrful x_0 și în fiecare muchie a lanțului apar două noduri etichetate cu x_0. Dacă u este nodul aflat la distanța cea mai mare de vîrful v, etichetat cu x_0, atunci toate nodurile etichetate cu x_0 se află în lanțul care unește u cu v.

Ne vom ocupa în continuare de unele proprietăți de închidere ale clasei limbajelor de tip 1. Amintim (v. teoremele II.2.1, II.2.5 și II.2.6) că această clasă de limbaje este închisă la reuniune, produs și închidere prin produs.

Definiția II.10.8. *O operație de substituție nesingulară generată de mulțimea de limbaje* $\{E_1, ..., E_r\}$ este o operație de substituție (v. def. II.4.3), pentru care $e \notin E_j$, $\forall j = \overline{1, r}$.

Teorema II.10.7. *Clasa limbajelor de tip 1 este închisă la operații de substituție nesingulare.*

Demonstrație. Fie $G = (I_N, \{i_1, i_2, ..., i_s\}, x_0; F)$ o gramatică de tip 1 și Φ_M o operație de substituție nesingulară generată de familia de limbaje $M = \{E_1, ..., E_n\}$, $E_j \in \mathcal{L}_1$, $j = \overline{1, n}$.

Fie $G_j = (I_{Nj}, I_{Tj}, x_{0j}; F_j)$ o gramatică de tip 1, care generează limbajul E_j, $j = \overline{1, n}$, $I_{Tj} \subseteq I_T$, $j = \overline{1, n}$.

Putem presupune că toate alfabetele neterminale sînt disjuncte două cîte două și că gramaticile care intervin au regulile de generare care conțin simboli terminali de forma $x \to i$, x fiind un simbol neterminal, iar i unul terminal (v. teorema II.10.4). Celelalte reguli de generare au forma $r' \to r''$, r', r'' fiind cuvinte alcătuite din simboli neterminali.

Mai putem avea, eventual, și reguli de forma $x \to e$.

Să construim acum o gramatică de tip 1 capabilă să genereze pe $\Phi_M(L(G))$.

În acest scop, fie $G' = (I'_N, I'_T, x_0; F')$, unde $I'_N = I_N \cup \bigcup_{j=1}^{n} I_{Nj} \cup \{\overline{x}, | x \in I_N\}$, $I'_T = \bigcup_{j=1}^{n} I_{Tj}$, iar F' constă din următoarele reguli de generare:

1) $\overline{x}_0 \to e$, dacă $(x_0 \to e) \in F$,

2) $\overline{x}_1 r_1 \to x_2 r_2$ și $x_1 r_1 \to x_2 r_2$ dacă $(x_1 r_1 \to x_2 r_2) \in F$,

3) $\bar{x} \to x_{0j}$, dacă $(x \to i_j) \in F$,

4) $ix \to ix_0$, pentru orice $i \in I'_T$, dacă $(x \to i_j) \in F$,

5) regulile de generare din $F_1, F_2, \ldots, F_n$.

Observăm că gramatica G' este de tip 1, întrucît toate regulile de generare satisfac condițiile impuse acestui tip de gramatică.

Fie $p \in \Phi_M(L(G))$. Există $q \in L(G)$, $q = i_{k_1} \ldots i_{k_m}$ și $q_1, \ldots, q_m$ cu $q_h \in E_{k_h}$, astfel încît $p = q_1 \ldots q_m$. Avem deducțiile $x_0 \overset{\bullet}{\underset{G}{\Rightarrow}} q$, $x_{0j} \overset{\bullet}{\underset{G_j}{\Rightarrow}} q_1$.

Să presupunem că deducția $x_0 \overset{\bullet}{\underset{G}{\Rightarrow}} q$ are aspectul $x_0 \overset{\bullet}{\underset{G}{\Rightarrow}} r \overset{\bullet}{\underset{G}{\Rightarrow}} q$, unde $r = x_{k_1} \ldots x_{k_m} \in I_N^*$, iar regulile de generare care participă în deducția $r \overset{\bullet}{\underset{G}{\Rightarrow}} q$ sînt de forma $x_{k_h} \to i_{k_h}$, $h = \overline{1, m}$.

În gramatica G, avem deducția $\bar{x}_0 \overset{\bullet}{\underset{G'}{\Rightarrow}} \bar{x}_{k_1} x_{k_2} \ldots x_{k_m} \underset{G'}{\Rightarrow} x_{0k_1} x_{k_2} \ldots x_{k_m} \overset{\bullet}{\underset{G'}{\Rightarrow}}$

$\Rightarrow q_1 x_{k_2} \ldots x_{k_m} \underset{G'}{\Rightarrow} q_1 x_{0k_2} x_{k_3} \ldots x_{k_m} \overset{\bullet}{\underset{G'}{\Rightarrow}} q_1 q_2 x_{k_3}, \ldots, x_{k_m} \underset{G'}{\Rightarrow} q_1 q_2 x_{0k_3} x_{k_4}, \ldots x_{k_m} \overset{\bullet}{\underset{G'}{\Rightarrow}}$

$\overset{\bullet}{\Rightarrow} q_1 q_2, \ldots, q_m = p$, deci $p \in L(G')$.

Invers, fie $p \in L(G')$. Există deducția $x_0 \overset{\bullet}{\underset{G'}{\Rightarrow}} p$, în care putem folosi reguli de generare de la punctul 5) și reguli de generare de la punctele 1)–4). Întrucît simbolul inițial al gramaticii G' este $\bar{x}_0$ în orice deducție în G' apar în mod necesar reguli de la punctele 1)–4).

Aplicarea unei reguli de generare de la punctul 5) se poate face numai după aplicarea prealabilă a unei reguli de forma 2) și 3) sau 4).

Întrucît regulile de generare de la punctele 2)–4) provin exclusiv din regulile de generare din gramatica G, avem corespunzător în gramatica G deducția $x_0 \overset{\bullet}{\underset{G}{\Rightarrow}} q = i_{k_1} i_{k_2} \ldots i_{k_m}$.

Această deducție poate fi detaliată astfel:

$$x_0 \overset{\bullet}{\underset{G}{\Rightarrow}} x_{k_1} \ldots x_{k_m} \overset{\bullet}{\Rightarrow} i_{k_1} \ldots i_{k_m} = q.$$

Prin urmare, deducția $\bar{x}_0 \overset{\bullet}{\underset{G'}{\Rightarrow}} p$ poate fi scrisă și

$$\bar{x}_0 \overset{\bullet}{\underset{G'}{\Rightarrow}} \bar{x}_{k_1} x_{k_2} \ldots x_{k_m} \underset{G'}{\Rightarrow} x_{0k_1} x_{k_2} \ldots x_{k_m} \overset{\bullet}{\underset{G'}{\Rightarrow}} q_{k_1} x_{k_2} \ldots$$

$$x_{k_m} \underset{G'}{\Rightarrow} q_{k_1} x_{0k_2} \ldots x_{k_m} \overset{\bullet}{\underset{G'}{\Rightarrow}} q_{k_1} \ldots q_{k_m} = p, \text{ unde } q_{k_j} \in E_{k_j},\ j = \overline{1, m}.$$

Astfel $p \in \Phi_M(L(G))$, cuvîntul p fiind obținut prin operația de substituție din cuvîntul $q \in L(G)$: ■

Observația II.10.6. Condiția de nesingularitate impusă operației de substituție asigură că gramatica G' este de tipul 1. Ea se dovedește a fi esențială, întrucît, așa cum vom arăta ulterior (v. observația II.15.1), dacă se renunță la ea, se poate ieși din clasa limbajelor de tip 1.

Consecința II.10.3. Clasa limbajelor de tip 1 este închisă la endomorfismele nesingulare ale semigrupului liber I^* (unde prin endomorfism nesingular, înțelegem un endomorfism pentru care $h(i) \neq e$, $\forall i \in I$).

Pentru a dovedi aceasta, este suficient să luăm $E_{j_k} = \{\varphi(i_{j_k})\}$. În acest caz limbajele E_{j_k} fiind finite, sînt de tip 3, prin urmare și de tip 1.

Reamintim necesitatea demonstrării incluziunii $\mathcal{L}_1 \supset \mathcal{L}_2$ (v. observația II.2.2). Avem astfel:

Teorema II.10.8. *Are loc incluziunea strictă*: $\mathcal{L}_1 \supset \mathcal{L}_2$.

Demonstrație. În baza teoremei II.5.2, pentru orice gramatică G de tip 2 există o gramatică G_1 expansivă, echivalentă, deci o gramatică de tip 1 echivalentă. Prin urmare, $\mathcal{L}_2 \subseteq \mathcal{L}_1$. Pentru a arăta că incluziunea este strictă, să considerăm gramatica de tip 1, $G = (\{x_0, x_1, y, y_1, y_2, z, z_1, z_2, z_3\}, \{i_x, i_y, i_z\}, F; x_0)$, unde F constă din:

$$x_0 \to x_0 z_1,\ x_0 \to x_1 z_1,\ x_1 \to i_x y_1,\ x_1 \to i_x x_1 y_1,$$
$$y_1 z_1 \to y_1 z_3,\ y_1 z_3 \to y z_3,\ y z_3 \to y z,$$
$$y_1 y \to y_1 y_2,\ y_1 y_2 \to y y_2,\ y y_2 \to y y_1,$$
$$z z_1 \to z_2 x_1,\ z_2 z_1 \to z_2 z,\ z_2 z \to z_1 z,$$
$$y \to i_y,\ z \to i_z.$$

Limbajul generat de această gramatică este:

(II.10.3) $L(G) = \{i_x^n i_y^n i_z^n \mid n \geqslant 1\}$, care nu este un lumbaj de tip 2, dar este un limbaj de tip 1.

Mai rămîne să verificăm egalitatea (II.10.3). Pentru aceasta vom observa că primele patru reguli de generare permit deducerea cuvintelor

$$x_0 \overset{*}{\underset{G}{\Rightarrow}} i_x^n y_1^n z_1^m,\ m, n \geqslant 1.$$

Folosind în continuare celelalte reguli de generare, putem ajunge la un cuvînt din I_T^*, dacă, și numai dacă, $m = n$. Astfel obținem $x_0 \overset{*}{\underset{G}{\Rightarrow}} i_x^n i_y^n i_z^n$.

Într-adevăr, avem $x_0 \overset{*}{\underset{G}{\Rightarrow}} i_x^n y_1^n z_1^n \Rightarrow i_x^n y_1^{n-1} y_1 z_3 z_1^{n-1} \Rightarrow$

$$\Rightarrow i_x^n y_1^{n-1} y z z_1^{n-1} \Rightarrow i_x^n y_1^{n-2} y_1 y_2 z z_1^{n-1} \Rightarrow i_x^n y_1^{n-2} y_1 y_2 z_2 z_1 z_1^{n-2} \Rightarrow$$
$$\Rightarrow i_x^n y_1^{n-2} y y_2 z_2 z_1 z_1^{n-2} \Rightarrow i_x^n y_1^{n-2} y y_2 z_2 z z_1^{n-2} \Rightarrow i_x^n y_1^{n-2} y y_1 z_2 z z_1^{n-2} \Rightarrow$$
$$\Rightarrow i_x^n y_1^{n-2} y y_1 z_1 z z_1^{n-2} \Rightarrow i_x^n y_1^{n-2} y^2 z^2 z_1^{n-2} \Rightarrow i_x^n y^n z^n \Rightarrow i_x^n i_y^n i_z^n.$$

Dacă am avea $m \neq n$, atunci am obține $x_0 \overset{*}{\Rightarrow} i_x^n i_y^n i_z^n z_1^{m-n}$, dacă $m > n$ sau $x_0 \overset{*}{\Rightarrow} i_x^n y_1^{n-m} i_y^n i_z^n$, dacă $n > m$ eliminarea simbolilor neterminali z_1 sau y_1 ne mai fiind posibilă.

§ II. 11. AUTOMATE LINIAR MĂRGINITE

Vom studia o clasă de automate care servesc la recunoașterea limbajelor de tip 1.

Definiția II.11.1. *Un automat liniar mărginit* este un 5-uplu $ALM = (I, S, f; s_0, \#)$, unde I este un alfabet finit, $\# \in I$ este numit *simbolul de marcaj*, S este o mulțime nevidă, finită de stări interne, $s_0 \in S$ este *starea inițială*, iar $f: S \times I \to \mathcal{P}(S \times I \times \{1, 0, -1\})$, astfel încît, dacă $(s', i, k) \in f(s, \#)$, atunci $i = \#$, iar dacă $(s', i, k) \in f(s, i)$, atunci $i = \#$, este *funcția de tranziție* a automatului.

Un ALM nu înlocuiește simbolul $\#$ cu vreun alt simbol din I și nu poate înscrie într-o celulă ocupată de un simbol din I simbolul $\#$.

Observația II.11.1. Un ALM lucrează în mod nedeterminist. Aflîndu-se în starea s și primind la intrare un simbol i (aflat pe banda de intrare (v. fig. II.11.1)) automatul trece într-o nouă stare s', deplasează eventual banda într-un sens sau altul și înscrie pe bandă un nou simbol i'. Automatul înscrie pe bandă simbolul $\#$ dacă, și numai dacă, primește la intrare simbolul $\#$.

Prin urmare, din punctul de vedere al activității de înscriere pe bandă a simbolilor din alfabetul $I \setminus \{\#\}$, introducerea simbolului $\#$ pune capăt acestei activități.

Dacă $(s', i', 1) \in f(s, i)$ automatul poate deplasa banda de intrare cu o celulă la stînga, după ce a înscris în celula din dreptul capului de citire-scriere simbolul i, dacă $(s', i', -1) \in (s, i)$ deplasarea se poate executa la dreapta, iar dacă $(s', i', 0) \in f(s, i)$ banda poate rămîne pe loc.

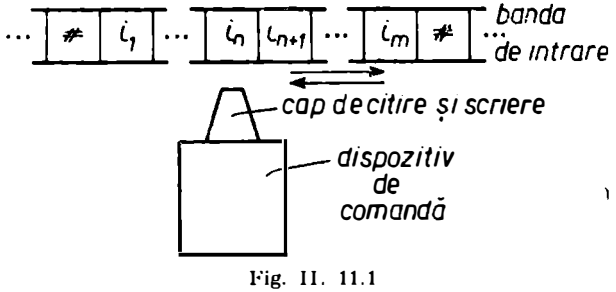

Fig. II. 11.1

Conținutnl benzii de intrare este reprezentat de un cuvînt $p \in I^*$ inscris între două marcaje #.

Activitatea automatului începe cu marcajul stîng al benzii de intrare și cu starea inițială s_0. Dacă printr-o serie de mișcări ale benzii și de tranziții succesive automatul *poate* ajunge în extremitatea dreaptă a benzii și trece într-o stare dintr-o mulțime dată de stări (numită mulțime de stări finale) cuvîntul aflat pe bandă este acceptat.

Definiția II.11.2. O *configurație instantanee* a unui $ALM = (I, S, f; s_0, \#)$ este un triplet $\delta = (p, s, q)$ unde $s \in S$, $p, q \in I^*$.

Dacă configurația instantanee este (p, s, q) automatul se află în starea s, banda conține cuvîntul $\# pq \#$, iar capul de citire-scriere se află în dreptul celulei care conține primul simbol din cuvîntul q.

Definiția II.11.3. *Configurația instantanee* $\delta_1 = (p_1, s_1, q_1)$ *precede imediat configurația instantanee* $\delta_2 = (p_2, s_2, q_2)$, dacă una din următoarele condiții este satisfăcută:

1) $q_1 = iq$ și $q_2 = i'q$, iar $(s_2, i', 0) \in f(s_1, i)$

2) $q_1 = iq$ și $p_2 = p_1 i'$, iar $(s_2, i', 1) \in f(s_1, i)$

3) $p_1 = pi_1$, $q_1 = i_1' q$ și $p_2 = p$, $q_2 = i_1 i_2 q$, iar $(s_2, i_2, -1) \in f(s_1, i_1')$.

Vom nota aceasta prin:

În cazul 1) automatul găsindu-se in starea s_1 și citind de pe banda de intrare simbolul i poate trece în starea s_2 și înscrie pe bandă simbolul i' în locul lui i și nu deplasează banda.

În cazul 2) automatul găsindu-se în starea s_1 și citind de pe banda de intrare simbolul i trece în starea s_2 și înscrie pe bandă simbolul i' în locul lui i, deplasează banda cu o celulă, cuvîntul aflat la stînga capului de citire-scriere, devenind $p_1 i'$. În sfîrșit, în cazul 3) automatul aflat în starea s_1 primind la intrare simbolul i_1' trece în starea s_2, înscrie în locul simbolului i_1' simbloul i_2 și mișcă banda la dreapta cu o celulă.

Definiția II.11.4. *Configurația instantanee* δ *precede configurația instantanee* δ', dacă există un șir finit de configurații instantanee

$$\delta = \delta_0, \delta_1, \ldots, \delta_n = \delta' \text{ cu } \delta_j \vdash \delta_{j+1}, \; j = \overline{0, n-1}.$$

Vom nota aceasta $\delta \overset{*}{\vdash} \delta'$. Șirul $\delta = \delta_0 \vdash \delta_1 \vdash \ldots \vdash \delta_n = \delta'$ îl vom numi *calcul* al configurației δ' pornind de la configurația δ, iar n va fi numit *lungimea calculului*.

Remarcăm că pentru fiecare configurație a unui calcul, lungimea cuvîntului conținut pe bandă rămîne neschimbată.

Definiția II.11.5. *Limbajul acceptat de un* ALM *prin mulțimea de stări finale* S_1 este:

$$L(ALM) = \{p \mid p \in I^*, (e, s_0, p) \overset{*}{\vdash} (q, s_1, e), s_1 \in S_1\}.$$

Definiția II.11.6. *Limbajul pe alfabetul* $I_1 \subseteq I$ *acceptat de un* $ALM = (I, S, f; s_0, \#)$ *prim mulțimea de stări finale* S_1 *este:*

$$L(ALM, I_1) = \{p \mid p \in I_1^*, (e, s_0, p) \overset{\bullet}{\vdash} (q, s_1, e), s_1 \in S_1, q \in I^*\}.$$

Evident $L(ALM, I) = L(ALM)$.

Teorema II.11.1. *Pentru orice limbaj* $E \subseteq I_T^*$ *de tip 1 există un automat liniar mărginit* ALM *astfel încît* $E = L(ALM, I_T)$.

Demonstrație. Conform teoremei II.10.6, putem presupune că limbajul $E \setminus \{e\}$ este generat de o gramatică liniar mărginită $G = (I_N, I_T, x_0; F)$ (v. definiția II.10.6), adică $(r_1 \to r_2) \in F$ dacă și numai dacă apare numai una din următoarele situații:

1) $r_1 = x_0$, $r_2 = x_0 x$, $x \in I_N$
2) $r_1 = x_1 x_2$, $r_2 = z_1 z_2$, $x_1, x_2, z_1, z_2 \in I_N$
3) $r_1 = x_1$, $r_2 = y$, $y \in I_N \cup I_T$, $x_1 \in I_N$, $x_0 \notin \{y, z_1, z_2, x\}$

Avem $e \in E$ dacă și numai dacă, $(x_0 \to e) \in F$.

Să considerăm automatul liniar mărginit

$$ALM = (I_N \cup I_T \cup \{\overset{\vdash}{\sqcup}, \#\}, S, f; s_0, \#).$$

unde $\overset{\vdash}{\sqcup}, \# \notin I_N \cup I_T$, $S = \{s_0, s_1, s_2, s_3, s_4\} \cup \{s_{x_j} \mid (x_j x \to z_1 z_2) \in F)\}$.

Mulțimea de stări finale este $\{s_3\}$, iar funcția de trecere este dată prin:

(II.11.1) $(s_0, \#, 1) = f(s_0, \#)$,

(II.11.2) $(s_4, \#, 1) = f(s_0, \#)$, dacă $e \in E$,

(II.11.3) $\{(s_0, y, 1), (s_0, y, -1)\} = f(s_0, y)$, pentru orice $y \in I_N \cup I_T$,

(II.11.4) $(s_0, x_1, 0) = f(s_0, y)$, dacă $(x_1 \to y) \in F$,

(II.11.5) $(s_{x_1}, x_1, 1) = f(s_0, z_1)$, dacă $(x_1 x_2 \to z_1 z_2) \in F$,

(II.11.6) $(s_0, x_2, 0) = f(s_{x_1}, z_2)$, dacă $(x_1 x_2 \to z_1 z_2) \in F$,

(II.11.7) $(s_1, x_0, -1) = f(s_0, x_0)$,

(II.11.8) $(s_2, \#, 1) = f(s_1, \#)$,

(II.11.9) $(s_2, \overset{\vdash}{\sqcup}, 1,) = f(s_1, \overset{\vdash}{\sqcup})$,

(II.11.10) $(s_3, \overset{\vdash}{\sqcup}, 1) = f(s_2, x_0)$,

(II.11.11) $(s_0, x_0, 0) = f(s_3, x_1)$, dacă $(x_0 \to x_0 x_1) \in F$,

(II.11.12) $(s_3, \#, 1) = f(s_3, \#)$,

(II.11.13) $(s_3, \#, 1) = f(s_4, \#)$

Pentru toate situațiile care nu apar în lista de mai sus, $f(s, y) = \varnothing$.

Relațiile (II.11.1–2) permit verificarea inițială a faptului dacă $e \in E$. Într-adevăr, să presupunem că $e \in E$. Dacă conținutul inițial al benzii este $\# \#$, în primul pas capul de citire-scriere lasă primul simbol „#" neschim-

bat, banda se deplasează la stînga și automatul trece în starea s_4. Apoi, conform relației (II.11.13), automatul trece în starea s_3 și marcajul drept trece în stînga capului de citire-scriere, deci $e \in L(ALM, I_T)$.

Invers, dacă $e \in L(ALM)$ aceasta înseamnă că există calculul:

$$(e, s_0, \# \#) \vdash (\#, s_4, \#) \vdash (\# \#, s_3, e), \text{ ceea ce implică}$$
$$(x_0 \to e) \in F, \text{ adică } e \in L(G) = E.$$

Relația (II.11.3) permite automatului să deplaseze banda într-un sens sau într-altul, menținîndu-și starea s_0 în căutarea unei situații potrivite pentru continuarea calculului.

Relațiile (II.11.4 – 6) reproduc respectiv regulile de generare din F, înlocuind simbolii din $I_N \cup I_T$ care apar în partea dreaptă a regulilor de generare prin simbolii corespunzători din partea stîngă.

Vom clarifica rolul jucat de relațiile (II.11.7 – 13) prin examinarea modului în care automatul recunoaște cuvîntul $p = i_1 i_2 i_1 i_1 i_2 i_2$ care aparține limbajului generat de gramatica liniar-mărginită considerată în exemplul II.10.1.

Configurațiile succesive sînt reprezentate în figura II.11.2 prin indicarea poziției relative a benzii și capului de citire-scriere.

Se observă deci că relațiile (II.11.7 – 13) sînt legate de regulile de generare de forma $x_0 \to x_0 x_1$, $x_1 \neq x_0$, $x_1 \in I_N$ în cazul în care $x_0 x_1$ se află în bandă în contextul $\# \overset{\vdash}{\sqcup}{}^h x_0 x_1 \ldots \#$ pentru $h \geqslant 0$. Cuvîntul $\# \overset{\vdash}{\sqcup}{}^h x_0 x_1 \ldots \#$ este înlocuit prin cuvîntul $\# \overset{\vdash}{\sqcup}{}^{h+1} x_0 \ldots \#$, pentru fiecare utilizare a unei reguli de generare de forma $x_0 \to x_0 x_1$, apărînd pe bandă un simbol $\overset{\vdash}{\sqcup}$.

Tot aceleași relații acționează și atunci cînd pe bandă se află cuvîntul $\# \overset{\vdash}{\sqcup}{}^n x_0 \#$ pentru $n \geqslant 0$, situație în care automatul iese prin capătul drept al benzii și intră în starea finală s_3.

Prin urmare automatul acceptă acele cuvinte din I_T^* care sînt generate în gramatica G, simulînd orice hiperarbore de deducție din gramatica liniar-mărginită G, pronind de la nodurile terminale ale hiperarborelui spre vîrf. Avem astfel $L(G) \subseteq L(ALM, I_T)$.

Să presupunem acum că $p \neq e$ este un cuvînt acceptat de automatul ALM, adică există un calcul în ALM:

$$(e, s_0, p) \overset{\bullet}{\vdash} (q, s_3, e),$$

unde s_3 este starea finală a automatului, adică la sfîrșitul calculului capul de citire-scriere trece în dreapta marcajului drept #, iar automatul trece în starea s_3, ceea ce se poate face folosind numai relația (II.11.12). Aceasta înseamnă că în pasul precedent capul de citire-scriere s-a aflat în dreptul marcajului final „#", iar automatul a fost în starea s_3, situația la care s-a putut ajunge numai prin utilizarea relației (II.11.10).

Continuînd în același mod, ținînd seama de relațiile (II.11.1 – 13), găsim hiperarborele de deducție, care ne permite să reconstituim deducția cuvîntului p în gramatica G, deci $p \in L(G)$. Cazul $p = e$ a fost tratat mai sus, deci $L(ALM, I_T) \subseteq L(G)$. ■

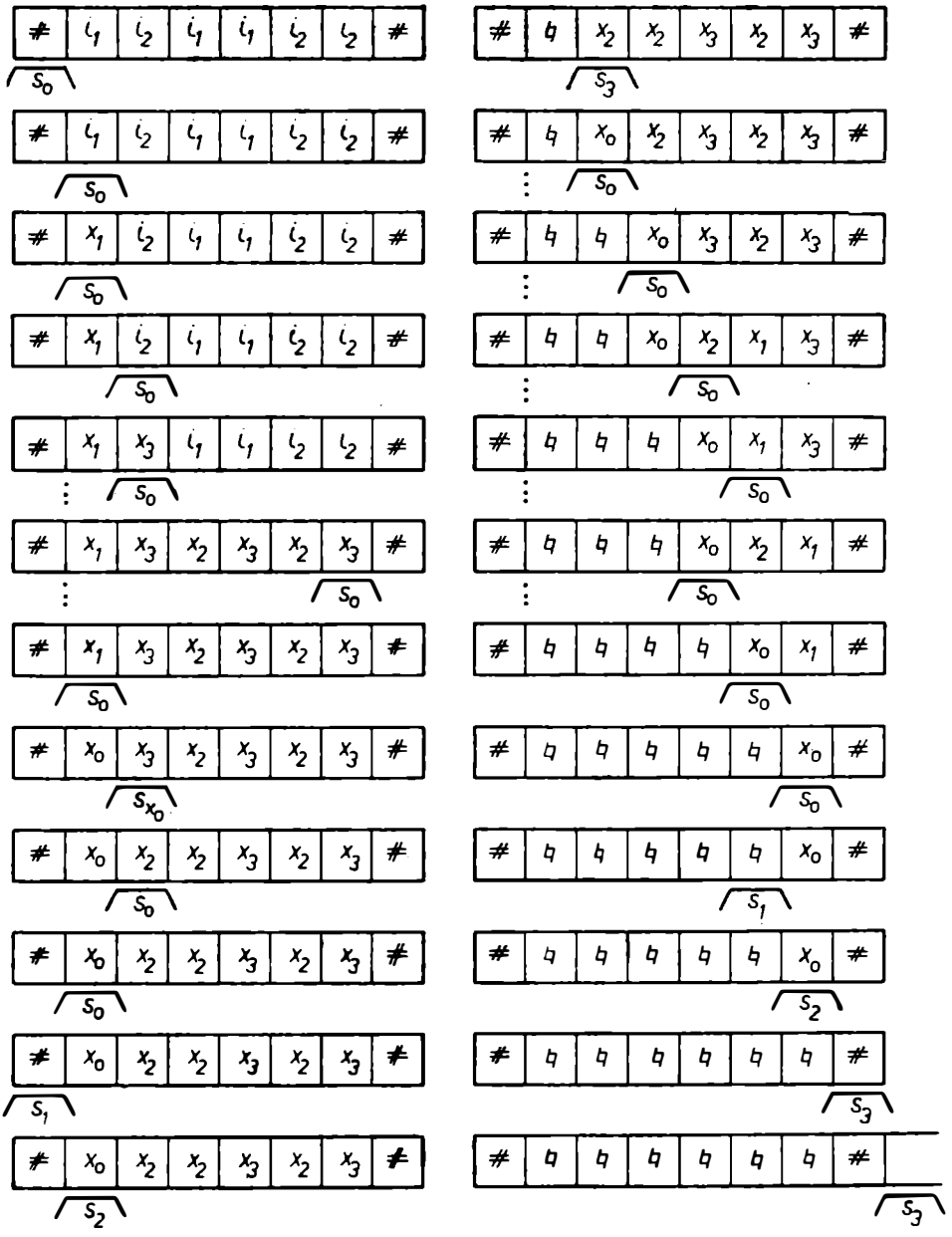

Fig. II. 11.2.

Teorema II.11.2. *Limbajul L (ALM, I_1) acceptat de automatul liniar mărginit $ALM = (I, S, f; s_0, \#)$ prin mulțimea de stări finale $S_1 \subseteq S$ și $I_1 \subseteq I$ este de tip* **1**.

Demonstrație. Fie $S_1' = \{s_1' \mid s_1' \in S, \exists s_1', \ldots, s_k', k \geqslant 1,$

$(s_{j+1}', \#, 0) \in f(s_j', \#), j = \overline{1, k-1}$ și $(s', \#, 1) \in f(s_k', \#), s' \in S_1\}$,

adică S_1' este mulțimea stărilor automatului cu proprietatea că automatul, găsindu-se într-o stare din S_1' și având la intrare simbolul $\#$, evoluează printr-un șir de stări fără a modifica conținutul celulei de pe bandă și fără a mișca banda, iar în ultimul pas deplasează banda la stînga, trecînd într-o stare finală din S_1.

Să considerăm gramatica $G = (I_N, I_1, x_0; F)$ unde I_N este dat de:

1) pentru $s \in S$, $i \in I$, I_N conține simbolii x_{si}, $\#x_{si}$, $x_{si}\#$, $\#x_{si}\#$,

2) pentru $i \in I$, I_N conține simbolii $\#i$, $i\#$

3) $I \setminus I_1 \subset I_N$

4) I_N conține simbolii x_0 și x_1, $x_0, x_1 \notin I$.

Mulțimea F constă din următoarele reguli de generare:

1) $x_1 \to \# i$, $x_1 \to x_1 i$, $\forall i \in I$

2) $x_0 \to \# x_{si}\#$, $x_0 \to x_1 x_{si}\#$, dacă $(s_1', y, 1) \in f(s, i)$ cu $y \in I, s_1' \in S_1'$

3) $x_{s'z}y \to zx_{si}$, $\#x_{s'z}\ y \to \#zx_{si}$, $x_{s'z}y\# \to zx_{si}\#$, $\#x_{s'z}\ y\# \to \#zx_{si}\#$, dacă $(s', y, -1) \in f(s, i)$ și $z \in I$.

4) $yx_{s'z} \to x_{si}z$, $\#yx_{s'z} \to \#x_{si}z$, $yx_{s'z}\# \to x_{si}z\#$, $\#yx_{s'z}\# \to \#x_{si}z\#$, dacă $(s', y, 1) \in f(s, i)$ și $z \in I$.

5) $x_{s'y} \to x_{si}$, $\#x_{s'y} \to \#x_{si}$, $x_{s'y}\# \to x_{si}\#$, $\#x_{s'y}\# \to \#x_{si}\#$, dacă $(s', y, 0) \in f(s, i)$.

6) $x_{s'y}\# \to x_{si}\#$, $\#x_{s'y}\# \to \#x_{si}\#$, dacă există $\{s_1'', \ldots, s_k''\} \subseteq S$, $k \geqslant 1$, astfel ca $(s_1'', y, 1) \in f(s, i)$, $(s_{j+1}'', \#, 0) \in f(s_j'', \#)$, $j = \overline{1, k-1}$, $(s', \#, -1) \in f(s_k'', \#)$.

7) $\#x_{s'y} \to \#x_{si}$, $\#x_{s'y}\# \to \#x_{si}\#$, dacă există $\{s_1'' \ldots s_k''\} \subseteq S$, $k \geqslant 1$, astfel ca $(s_1'', y, -1) \in f(s, i)$, $(s_{j+1}'', \#, 0) \in f(s_j'', \#)$, $j = \overline{1, k-1}$, $(s', \#, 1) \in f(s_k'', \#)$.

8) $i\# \to i$, $\forall i \in I$

9) $\#x_{si} \to i$, $\#x_{si}\# \to i$ dacă $(s, \#, 1) \in f(s_0, \#)$. Vom presupune că $f(s_0, \#)$ conține numai tripletele de forma $(s', \#, 1)$, deoarece oricum plecînd de la marcajul $\#$, banda se deplasează la stînga.

10. $x_0 \to e$, dacă $e \in L(ALM, I_1)$.

Gramatica G expansivă definită aici generează un limbaj de tip 1, conform teoremei II.10.2. Fie E mulțimea cuvintelor p pentru care $x_0 \overset{\bullet}{\underset{G_{1-7}}{\Rightarrow}} p$ și p nu conține simbolii x_0 și x_1, unde cu G_{1-7} am indicat folosirea exclusivă a regulilor de generare 1) – 7).

Observăm că orice cuvînt $p \in E$ conține exact un simbol x_{si} (cu unul sau doi simboli #). Primul simbol din p este marcat la stînga cu „#", iar ultimul este marcat la dreapta cu #, iar simbolii intermediari nu sînt marcați.

Să considerăm aplicația $\varphi : E \to (S \cup I \cup \{\#\})^*$ dată prin :

$$\varphi(p) = \varphi(y_1 y_2 \dots y_n) = \varphi(y_1) \dots \varphi(y_n) \text{ și}$$

$$\varphi(y_j) = \begin{cases} si & \text{, dacă } y_j = x_{si} \\ \#si & \text{, dacă } y_j = \underset{si}{\#} \\ si\# & \text{, dacă } y_j = \underset{si}{\#} \\ \#si\# & \text{, dacă } y_j = \#\underset{si}{x}\# \\ y_j & \text{, în celelalte cazuri.} \end{cases}$$

Cuvîntul $\varphi(p)$ are aspectul $\varphi(p) = \# r_2 s r_2 \#$, $r_1, r_2 \in I^*$.

Vom atașa cuvîntului p configurația $\delta(p) = (r_1, s, r_2)$.

Pentru $p_1, p_2 \in E$ avem $p_1 \overset{\bullet}{\underset{G}{\Rightarrow}} p_2$, dacă, și numai dacă $\delta(p_2) \overset{\bullet}{\vdash} \delta(p_1)$, deoarece fiecărui pas din deducția $p_1 \overset{\bullet}{\Rightarrow} p_2$ îi corespunde un pas în calculul $\delta(p_2) \overset{\bullet}{\vdash} \delta(p_1)$ și reciproc, făcînd convenția că șirurile de configurații :

$$(q, s, i) \vdash (qy, s_1'', e) \vdash \dots \vdash (qy, s_k'', e)\ (q, s', y) \text{ și}$$

$(i, s, q) \vdash (e, s_1'', yq) \vdash \dots \vdash (e, s_k'', yq) \vdash (y, s', q)$ reprezintă un singur pas în calcul.

Regulile de generare descrise în 1) și 2) permit deducțiile $x_0 \overset{\bullet}{\Rightarrow} p$ numai pentru acele cuvinte p cu $\varphi(p) = \#q$ și #, $q \in I^*$, adică pentru acele cuvinte pentru care $\delta(p) = (q, s, i)$.

Aplicarea regulilor descrise în 1) și 2) presupune că automatul a trecut (deplasînd banda la stînga) într-o stare din mulțimea S_1'.

Folosirea regulilor 8) și 9) este singura modalitate de a elimina simbolii neterminali din cuvintele deduse.

Să arătăm că $L(G) = L(ALM, I_1)$. Fie $p \in L(G)$. Pentru simplitatea expunerii alegem o formă particulară a cuvîntului p, în cazul general procedeul de demonstrare fiind similar. Fie deci $p = i_8 i_7 i_6 i_1 \in I_1^*$ și deducția :

$$x_0 \overset{2)}{\Rightarrow} x_1 x^{\#}_{s i_1} \overset{1)}{\Rightarrow} x_1 x_2 x^{\#}_{s i_1} \overset{1)}{\Rightarrow} x_1 i_3 i_2 x^{\#}_{s i_1} \overset{4_3)}{\Rightarrow} x_1 i_3 x_{s_1} i_4 i^{\#}_1 \overset{1_1)}{\Rightarrow}$$

$$\# i_5 i_3 x_{s_1 i_4} i^{\#}_1 \overset{5_1)}{\Rightarrow} \# i_5 i_3 x_{s_2 i_6} i^{\#}_1 \overset{4_1)}{\Rightarrow} \# i_5 x_{s_3 i_7} i_6 i^{\#}_1 \overset{4_2)}{\Rightarrow} x_{s_4 i_8} i_7$$

$$i_6 i^{\#}_1 \overset{9_1)}{\Rightarrow} i_8 i_7 i_6 i^{\#}_1 \overset{8)}{\Rightarrow} i_8 i_7 i_6 i_1 .$$

Am pus în evidență regula de generare folosită în fiecare pas al deducției indicînd-o pe săgeata dublă.

De exemplu „$\overset{4_1)}{\Rightarrow}$" înseamnă aplicarea primei reguli de la punctul 4).

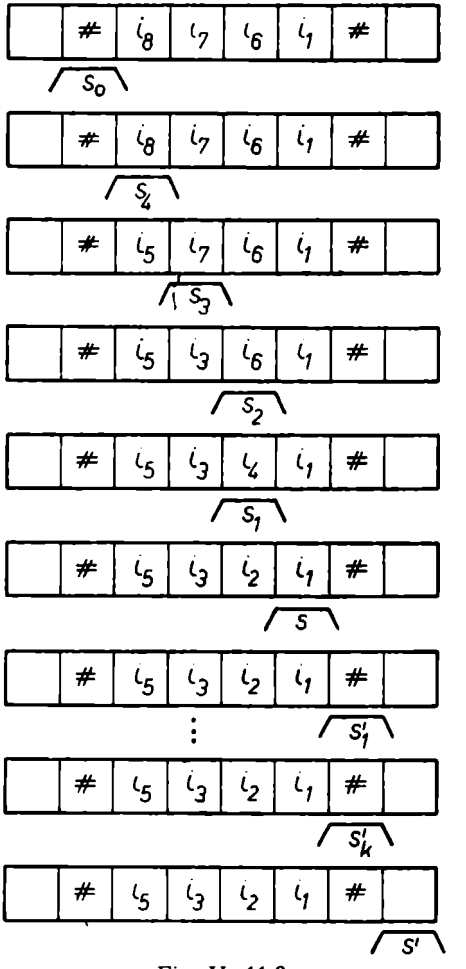

Fig. II. 11.3

Cuvîntului p îi corespunde calculul descris în fig. II.11.3. prin urmare $p \in L(ALM, I_1)$. Dacă $p = e$, există deducția $x_0 \overset{10)}{\Rightarrow} e$ și conform definiției gramaticii G, $p = e \in L(ALM, I_1)$. Deci $L(G) \subseteq L(ALM, I_1)$.

Invers, fie $p \in L(ALM, I_1)$. Atunci, ținînd seama de definiția gramaticii G, pentru un calcul al cuvîntului p există o deducție $x_0 \overset{*}{\underset{G}{\Rightarrow}} p$, care se obține parcurgînd în sens invers fazele calculului, deci $p \in L(G)$ și $L(ALM, I_1) \subseteq L(G)$. ■

Teorema II.11.3. *Clasa $\mathcal{L}_1$ este închisă la intersecție.*

Demonstrație. Fie L_1 și L_2 două limbaje de tip 1, $L_1 = L(ALM, I_1')$ și $L_2 = L(ALM_2, I_2')$, unde $ALM_1 = (I_1, S_1, f_1; s_{01}, \#)$, $ALM_2 = \{I_2, S_2, f_2; s_{02}, \#)$, limbajele L_1 și L_2 fiind reprezentate în aceste automate respectiv prin mulțimile de stări S_1' și S_2'.

Vom presupune $S_1 \cap S_2 = \varnothing$ și fie $I = I_1' \cap I_2'$.

Să considerăm automatul liniar mărginit $ALM = (I \cup (I_1 \times I_2), S_1 \cup S_2 \cup \{s_0, s_1, s_2, s_3\}, f; s_0, (\#, \#))$, unde $\{s_0, s_1, s_2, s_3\} \cap (S_1 \cup S_2) = \varnothing$, iar f este dat în tabelul II.11.1.

Tabelul II.11.1

Nr. crt.	(s, i)	$f(s, i)$
1	$(s_0, (\#, \#))$	$(s_1, (\#, \#), 1)$
2	(s_1, i), $\forall i \in I$	$(s_1, (i, i), 1)$
3	$(s_1, (\#, \#))$	$(s_2, (\#, \#), -1)$
4	$(s_2, (i, i))$, $\forall i \in I$	$(s_2, (i, i), -1)$
5	$(s_2, (\#, \#))$	$(s_{01}, (\#, \#), 0)$
6	$(s', (i_1, i_2))$, cu $(s', i_1) = f_1(\bar{s}_1, \bar{i}_1, k)$	$(\bar{s}_1, (\bar{i}_1, i_2), k)$, $i_2 \in I_2$
7	$(s', (\#, \#))$, cu $(s', \#) =$ $= f_1(s_1', \#, 1)$, $s_1' \in S_1'$	$(s_3, (\#, \#), -1)$
8	$(s_3, (i_1, i_2))$, $\forall (i_1, i_2) \in I_1 \times I_2$	$(s_3, (i_1, i_2), -1)$
9	$(s_3, (\#, \#))$	$(s_{02}, (\#, \#), 0)$
10	$(s'', (i_1, i_2))$ cu $(s'', i_2) =$ $= f_2(\bar{s}'', \bar{i}_2, k)$	$(s'', (i_1, \bar{i}_2), k)$, $i_1 \in I_1$

Să presupunem că în automatul ALM avem inițial pe bandă un șir de simboli din I. Conform posibilității descrise la punctul 2 al tabelului, automatul înlocuiește simbolii $i \in I$ cu perechi de forma (i, i), deplasînd banda spre stînga.

Acest proces se termină cînd se întîlneşte marcajul (#, #) din partea dreaptă. La întîlnirea acestui marcaj automatul trece în starea s_2 şi deplasează banda la dreapta pînă la întîlnirea marcajului (#, #) din partea stîngă, automatul fiind pus pe starea s_{01}.

În acest moment automatul începe să opereze asupra primilor simboli ai perechilor scrise pe bandă.

Cuvîntul p_1 alcătuit din aceşti simboli este din limbajul $L_1 = L(ALM, I_1')$ dacă, şi numai dacă, automatul ajunge în starea $s' \in S_1'$, în dreptul marcajului (#, #) din partea dreaptă. Deci, dacă $p \in L_1$ automatul trece în starea s_3 şi menţinîndu-şi această stare deplasează banda la dreapta, pînă cînd ajunge la marcajul (#, #) din stînga. Starea s_3 este înlocuită acum cu starea s_{02} şi se fac calculele care simulează cele executate de automatul ALM_2, avînd acelaşi cuvînt de intrare p.

Automatul ALM ajunge într-o stare din $S_2' \subseteq S_2$ dacă, şi numai dacă $p \in L_2$. În definitiv automatul ALM ajunge în mulţimea de stări finale S_2' dacă şi numai dacă $p \in L_1 \cap L_2$. Conchidem că limbajul reprezentat în automatul ALM prin mulţimea de stări S_2' este limbajul $L_1 \cap L_2$ şi deci acesta este de tip 1. ■

§ II. 12. MAŞINI TURING — DISPOZITIVE DE RECUNOAŞTERE A LIMBAJELOR

În acest paragraf vom studia dispozitivele capabile să accepte clasa limbajelor de tip 0.

O maşină Turing este alcătuită dintr-un *dispozitiv de comandă* care se poate afla într-o anume stare dintr-o mulţime de stări S, prevăzut cu un *cap de citire-scriere* care preia şi transmite informaţiile de pe o bandă (fig. II.12.1).

Această bandă este împărţită în celule, are o primă celulă în stînga şi este infinită la dreapta. Celulele pot conţine cîte un simbol dintr-un alfabet finit $I = \{i_1, ..., i_n\} \cup \{\mathbf{b}\}$. Vom presupune că în orice moment numai un număr finit de celule, începînd cu prima celulă, pot conţine simboli din $I \setminus \{\mathbf{b}\}$, restul celulelor conţinînd simbolul $\mathbf{b}$.

La un moment dat, capul de citire-scriere citind simbolul din celula din dreptul căreia se află, starea automatului se schimbă, se înscrie un simbol pe bandă în aceeaşi celulă, după care banda rămîne pe loc sau este deplasată la dreapta sau la stînga cu o celulă.

Descrierea de mai sus este formalizată în:

Definiţia II.12.1. O *maşină Turing* este un 5-uplu: $MT = (I, S, f; \mathbf{b}, s_0)$, unde I este un alfabet finit, S o mulţime finită, nevidă, numită *mulţimea*

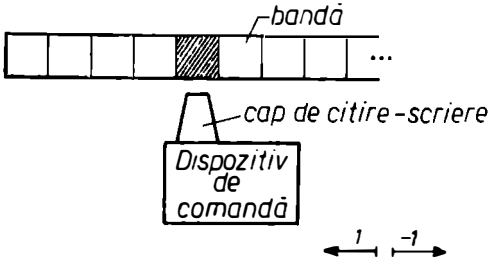

Fig. II. 12.1

stărilor, $f : (S \times I)' \to S \times \{I \setminus \{\mathbf{b}\}\} \times \{-1, 0, 1\}$ este *funcția de trecere*, $(S \times I)' \subseteq S \times I$ este mulțimea perechilor de forma (s, i) pentru care funcția f este definită, $\mathbf{b} \in I$ este *simbolul „blanc"*, iar s_0 este *starea inițială* a mașinii. Dacă $(s, i) \notin (S \times I)'$ perechea (s, i) este *o situație de oprire* pentru MT.

O mașină Turing nestaționară MTN este o MT pentru care $f : (S \times I)' \to S \times \{I \setminus \{\mathbf{b}\}\} \times \{-1, 1\}$.

Observația II.12.1. O MT (MTN) funcționează în mod determinist. Dacă $f(s, i) = (s', i', 1)$, atunci MT găsindu-se în starea s și citind simbolul i, trece în starea s', înlocuiește pe i cu simbolul i' și deplasează banda la *stînga* cu o celulă; dacă $f(s, i) = (s', i', 0)$, banda rămîne *pe loc*, iar dacă $f(s, i) = (s', i', -1)$, banda se deplasează la *dreapta* cu o celulă, automatul acționînd în rest la fel ca mai sus. Pentru MTN sînt posibile numai deplasări ale benzii la dreapta și la stînga, banda neputînd rămîne pe loc.

Definiția II.12.2. *O configurație instantanee* a unei MT este un triplet $\gamma = (s, p, n)$, unde $s \in S$ este starea curentă a mașinii, $p \in (I \setminus \{\mathbf{b}\})^*$ este conținutul porțiunii inițiale din bandă, pînă la primul simbol blanc, iar n este numărul de ordine al celulei în dreptul căreia se află capul de citire.

O configurație instantanee $\gamma_1 = (s_1, p_1, n_1)$ *precede imediat configurația instantanee* $\gamma_2 = (s_2, p_2, n_2)$ dacă, presupunînd că $p_1 = i_1 i_2 \dots i_n$, atunci avem una din situațiile:

1) $f(s_1, i_{n_1}) = (s_2, i, 1)$, $1 \leqslant n_1 \leqslant n$, $n_2 = n_1 + 1$, $p_2 = i_1 \dots i_{n_1 - 1} i i_{n_1 + 1} \dots i_n$;

2) $f(s_1, i_{n_1}) = (s_2, i, -1)$, $2 \leqslant n_1 \leqslant n$, $n_2 = n_1 - 1$, $p_2 = i_1 \dots i_{n_1 - 1} i i_{n_1 + 1} \dots i_n$;

3) $f(s_1, \mathbf{b}) = (s_2, i, 1)$, $n_1 = n + 1$, $n_2 = n_1 + 1$, $p_2 = p_1 i$;

4) $f(s_1, \mathbf{b}) = (s_2, i, -1)$, $n_1 = n + 1$, $n_2 = n_1 - 1$, $p_2 = p_1 i$;

5) $f(s, i_{n_1}) = (s_2, i, 0)$, $1 \leqslant n_1 \leqslant n$, $n_2 = n_1$, $p_2 = i_1 \dots i_{n_1 - 1} i i_{n_1 + 1} \dots i_n$;

6) $f(s, \mathbf{b}) = (s_2, i, 0)$, $n_1 = n + 1$, $n_2 = n_1$, $p_2 = p_1 i$;

Vom nota acest lucru prin:

$$\gamma_1 \vdash \gamma_2 .$$

Configurația instantanee γ' *precede configurația instantanee* γ'', dacă există un șir finit de configurații numit *calcul*: $\gamma' = \gamma_1 \vdash \gamma_2 \vdash \dots \vdash \gamma_n = \gamma''$, ceea ce se va nota cu:

$$\gamma' \overset{*}{\vdash} \gamma''$$

Definiția II.12.3. *Limbajul acceptat de o* $MT = (I, S, f; \mathbf{b}, s_0)$ *prin mulțimea de stări finale* $S_1 \subseteq S$ este:

$$L(MT) = \{p \mid p \in \{I \setminus \{\mathbf{b}\}\}^*, (s_0, p, 1) \overset{*}{\vdash} (s, q, n_1), s \in S_1, q \in I^*\}.$$

Observația II.12.2. Limbajul acceptat de MT este alcătuit din acele cuvinte, care plasate pe bandă, începînd cu prima celulă (automatul găsindu-se inițial în starea s_0), conduc la o stare din S_1, activitatea mașinii începînd cu prima celulă.

Observația II.12.3. În continuare vom presupune că, dacă $s \in S_1 \mapsto f(s, i) = (\ldots, 0)$ adică, dacă mașina intră într-una din stările finale, banda se oprește.

Exemplul II.12.1. Fie $MT_1 = (\{0, 1, x, y, b\}, (s_0, s_1, s_2, s_3, s_4, s_5\}, f; \mathbf{b}, s_0)$ unde f este dat în tabelul II.12.1.

Tabelul II.12.1

Nr. crt.	s	i	$f(s, i)$
I.1	s_0	0	$(s_1, x, 1)$
II.1	s_1	0	$(s_1, 0, 1)$
II.2	s_1	y	$(s_1, y, 1)$
II.3	s_1	1	$(s_2, y, -1)$
III.1	s_2	y	$(s_2, y, -1)$
III.2	s_2	x	$(s_3, x, 1)$
III.3	s_2	0	$(s_4, 0, -1)$
IV.1	s_4	0	$(s_4, 0, -1)$
IV.2	s_4	x	$(s_0, x, 1)$
V.1	s_3	y	$(s_3, y, 1)$
V.2	s_3	$\mathbf{b}$	$(s_5, y, 1)$

iar $(S \times I)' = \{(s_0, 0), (s_1, 0), (s_1, y), (s_1, 1), (s_2, y), (s_2, x), (s_2, 0), (s_4, 0)$ $(s_4, x), (s_3, y), (s_3, \mathbf{b})\}$.

Să presupunem că pe bandă este înscris cuvîntul 000111.

Configurațiile succesive prin care trece mașina sînt date în tabelul II.12.2.

Tabelul II.12.2

1	Tact	1	2	3	4	5	6
2	Configurație	$(s_0, \bar{0}00111, 1)$	$(s_1, x\bar{0}0111, 2)$	$(s_1, x0\bar{0}111, 3)$	$(s_1, x00\bar{1}11, 4)$	$(s_2, x0\bar{0}y11, 3)$	$(s_4, x\bar{0}0y11, 2)$
3	Regula folosită		I.1	II.1	II.1	II.3	III.3

1	7	8	9	10	11	12	13
2	$(s_4, \bar{x}00y11, 1)$	$(s_0, x\bar{0}0y11, 2)$	$(s_1, xx\bar{0}y11, 3)$	$(s_1, xx0\bar{y}11, 4)$	$(s_1, xx1\bar{y}11, 5)$	$(s_3, xx0\bar{y}y1, 4)$	$(s_2, xx\bar{0}yy1, 3)$
3	IV.1	IV.2	I.1	II.1	II.2	II.3	III.1

1	14	15	16	17	18	19	20
2	$(s_4, \bar{xx}0yy1, 2)$	$(s_0, xx\bar{0}yy1, 3$	$(s_1, xxx\bar{y}y1, 4)$	$(s_1, xxxy\bar{y}1, 5)$	$(s_1, xxxyy\bar{1}, 6)$	$(s_3, xxx\bar{y}yy, 5)$	$(s_2, \bar{x}xxyyy, 4)$
3	III.3	IV.2	I.1	II.2	II.2	II.3	III.1

1	21	22	23	24	25	26
2	$(s_2, \bar{xx}yyy, 3)$	$(s_3, xx\bar{x}yyy, 4)$	$(s_3, xxx\bar{y}yy, 5)$	$(s_3, xxxy\bar{y}yy, 6)$	$(s_3, xxxyyy\bar{b}, 7)$	$(s_5, xxxyyyy\bar{b}, 8)$
3	III.1	III.2	V.1	V.1	V.1	V.2

În tabelul II.12.2 s-a indicat prin supraliniere simbolul din celula din dreptul capului citire-scriere. Examinînd acest tablou, concludem că cuvîntul 000111 face parte din limbajul acceptat de mașina Turing dată, prin mulțimea de stări finale $S_1 = \{s_5\}$.

Exemplul II.12.2. Să considerăm $MT_2 = (\{0, 1, \mathbf{b}\}, \{s_0, s_1\} \times \{0, 1, \mathbf{b}\}, f; \mathbf{b}, (s_0, \mathbf{b}))$ avînd mulțimea de stări finale $\{(s_1, \mathbf{b})\}$. Funcția f este dată de tabelul II.12.3 :

Tabelul II.12.3

Nr. crt.	s	i	$f(s, i)$
I.1	$(s_0, \mathbf{b})$	0	$((s_1, 0), 0, 1)$
I.2	$(s_0, \mathbf{b})$	1	$((s_1, 1), 1, 1)$
II.1	$(s_1, 0)$	1	$((s_1, 0), 1, 1)$
II.2	$(s_1, 1)$	0	$((s_1, 1), 0, 1)$
III.1	$(s_1, 0)$	$\mathbf{b}$	$((s_1, \mathbf{b}), 0, -1)$
III.2	$(s_1, 1)$	$\mathbf{b}$	$((s_1, \mathbf{b}), 0, -1)$

Mulțimea stărilor acestei mașini este compusă din perechi de forma (s, i). Rolul componentei a doua din o stare este memorarea simbolului de intrare citit de pe o bandă atunci cînd mașina se află în starea $(s_0, \mathbf{b})$.

Trecerea în starea finală $(s_1, \mathbf{b})$ se produce atunci cînd mașina aflîndu-se într-una din stările $(s_1, 0)$, $(s_1, 1)$ citește semnul $\mathbf{b}$ de pe bandă. În această situație se produce memorarea simbolului $\mathbf{b}$, înlocuirea lui cu 0 respectiv 1 și deplasarea benzii la stînga.

Limbajul recunoscut de această mașină este $L = \{01^n \mid n \in N\} \cup \{10^m \mid m \in N\}$.

Definiția II.12.4. *O mașină Turing bilateral infinită* este un 5-uplu $MTBI = (I, S, f; \mathbf{b}, s_0)$, în care I, S, f, $\mathbf{b}$, s_0 au semnificațiile din definiția II.12.1 iar banda mașinii este nemărginită în ambele sensuri.

Pe bandă este înscris un cuvînt $p \in (I \setminus \{\mathbf{b}\})^*$, iar în celulele situate la dreapta și la stînga cuvîntului se află blancuri.

Definiția II.12.5. *O configurație instantanee* a unei $MTBI$ este un triplet $\gamma = (s, p, n)$, unde $s \in S$ este starea curentă a mașinii, $p \in (I \setminus \{\mathbf{b}\})^*$ este cuvîntul înscris pe bandă, iar n este numărul de ordine al celulei în dreptul căreia se află capul de citire-scriere, numerotarea începînd cu celula în care se găsește primul simbol din cuvîntul p.

Fie γ_1, γ_2 două configurații instantanee. Sensul expresiilor $\gamma_1 \vdash \gamma_2$ și $\gamma_1 \overset{*}{\vdash} \gamma_2$ este același ca în definiția II.12.2 dacă $n > 1$.

Dacă $f(s, i) = (s_1, i_1, -1)$, atunci $(s, ip, 1) \vdash (s_1, i_2\, p, 0)$, dacă $f(s, \mathbf{b}) = (s_1, i_1, 1)$, atunci $(s, p, 0) \vdash (s_1, i_1\, p, 2)$ și, în fine, dacă $f(s, \mathbf{b}) = (s_1, i_1, -1)$, atunci $(s, p, 0) \vdash (s_1, i_1\, p, 0)$, cu $i_1 \neq \mathbf{b}$.

Definiția II.12.6. *Limbajul acceptat de o* $MTBI = (I, S, f; \mathbf{b}, s_0)$ *prin mulțimea de stări finale* $S_1 \subseteq S$ este:

$$L(MTBI) = \{p \mid p \in (I \setminus \{\mathbf{b}\})^*,\ (s_0, p, m) \overset{\bullet}{\vdash} (s, q, n),\ s \in S_1,\ q \in I^*\}.$$

Mașinile Turing bilateral infinite au aceeași capacitate de a accepta limbaje ca și mașinile Turing, ceea ce se precizează în:

Teorema II.12.1. *Un limbaj este acceptat de o* $MTBI$ *dacă, și numai dacă, este acceptat de o* MT.

Demonstrație. Deoarece o MT poate fi privită ca o $MTBI$ la care capul de citire-scriere nu parcurge niciodată partea infinită din stînga cuvîntului scris pe bandă, este evident că un limbaj acceptat de o MT este acceptat de o $MTBI$.

Fie acum E un limbaj acceptat de mașina Turing bilateral infinită $MTBI = (I, S, f; \mathbf{b}, s_0)$ prin mulțimea stărilor finale S'. Construim o mașină Turing $MT = (I_1, S_1, f_1; (\mathbf{b}, \mathbf{b}), s_{01})$ care va accepta același limbaj. Alfabetul de intrare $I_1 = I \times (I \cup \{!\})$ este alcătuit din perechi de simboluri. Ne putem imagina că banda de intrare a mașinii MT are două piste care conțin cîte una din componentele fiecărei perechi $(i_1, i_2) \in I_1$. Pe prima pistă se află înscris conținutul benzii mașinii $MTBI$, începînd cu celula din dreptul capului de citire-scriere și continuînd la dreapta, iar a doua pistă va conține simbolul ! în prima celulă urmat de conținutul benzii mașinii $MTBI$ situat la stînga capului de citire-scriere, scris în ordine inversă. De exemplu, dacă banda mașinii $MTBI$ are conținutul din figura II.12.1a, atunci banda mașinii MT are aspectul din figura II.12.1.b. Vom lua $S_1 = \{(s, \uparrow), (s, \downarrow) \mid s \in S\} \cup \{s_{01}\}$ și vom nota în loc de $(i_1, \mathbf{b})$ pur și simplu i_1. Funcția de tranziție f este dată de tabelul II.12.4.

Dacă în momentul inițial $MTBI$ deplasează banda la dreapta înscriind în locul simbolului de intrare i simbolul i' și trecînd în starea s', MT înscrie pe banda sa cu două piste $(i', !)$ și trece în starea $(s', \uparrow)$, ceea ce indică faptul că MT va urmări simbolii situați pe pista superioară a benzii.

Cazurile III.1 și III.2 arată că mașina Turing MT operează pe pista superioară aceleiași modificări ca acelea efectuate de $MTBI$ în partea dreaptă a simbolului inițial.

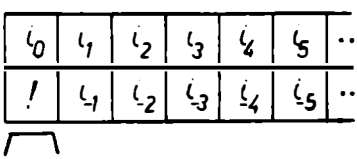

Fig. II. 12.1 a

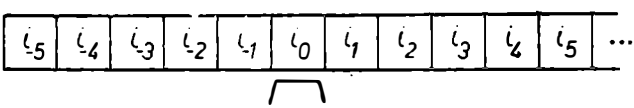

Fig. II. 12.1b

Tabelul II.12.4

Nr. crt.	Evoluția mașinii *MTBI*	Evoluția mașinii *MT*
I	$f(s_0, i) = (s', i', 1)$	$f_1(s_{01}, (i, \mathbf{b})) = ((s', \uparrow), (i', !), 1)$
II	$f(s_0, i) = (s', i', -1)$	$f_1(s_{01}, (i, \mathbf{b})) = ((s', \downarrow), (i', !), 1)$
III.1	$f(s, i) = (s', i', 1)$	$f_1((s, \uparrow), (i, i'')) = ((s', \uparrow), (i', i''), 1), i'' \neq !$
III.2	$f(s, i) = (s', i', -1)$	$f_1((s, \uparrow), (i, i'')) = ((s', \uparrow), (i', i''), -1), i'' \neq !$
IV.1	$f(s, i'') = (s', i', 1)$	$f_1((s, \downarrow), (i, i'')) = ((s', \downarrow), (i, i'), -1)$
IV.2	$f(s, i'') = (s', i', -1)$	$f_1((s, \downarrow), (i, i'')) = ((s', \downarrow), (i, i'), 1)$
V.1	$f(s, i) = (s', i'', 1)$	$f_1((s, \uparrow), (i, !)) = f_1((s, \downarrow), (i, !)) = ((s', \uparrow), (i'', !), 1)$
V.2	$f(s, i) = f(s', i'', -1)$	$f_1((s, \uparrow), (i, !)) = f_1((s, \downarrow), (i, !)) = ((s', \downarrow), (i'', !), 1)$

Situația este similară pentru deplasările la stînga ale benzii mașinii $MTBI$, în acest caz mașina MT prelucrînd simbolii de pe pista inferioară și deplasîndu-se în sens invers (v. cazurile II și IV.1.,2).

Dacă la un moment dat în dreptul capului de citire-scriere al mașinii $MTBI$ se află iar celula inițială, atunci mașina MT prelucrează simbolul de pe pista superioară sau inferioară, după sensul în care se mișcă banda mașinii $MTBI$, banda lui MT mișcîndu-se totdeauna la dreapta.

Mulțimea de stări finale ale lui MT este: $S_1' = \{(s, \uparrow), (s, \downarrow) \mid s \in S'\}$.

Să arătăm că limbajul acceptat de $MTBI$ prin mulțimea de stări S_1' coincide cu limbajul acceptat de mașina MT prin mulțimea de stări S'.

Fie $p \in L(MTBI)$. Avem calculul în $MTBI$: $(s_0, p, n) \overset{*}{\vdash} (s, q, n_1)$ cu $s \in S_1$, $q \in I^*$.

Pentru simplitatea expunerii vom presupune că $p = i_1 i_2 i_3 i_4 i_5$ și vom detalia calculul astfel: $(s_0, i_1 i_2 i_3 i_4 i_5, 4) \vdash (s_1 i_1 i_2 i_3 i_6 i_5, 3) \vdash (s_2, i_1 i_2 i_7 i_6 i_5$ $2) \vdash (s_3, i_1 i_2 i_7 i_8 i_5, 3) \vdash (s_4, i_1 i_2 i_8 i_6 i_5, 4) \vdash (s, i_1 i_2 i_8 i_9 i_5, 5)$ cu $s \in S_1$. Diversele situații prin care trece $MTBI$ sînt descrise în figura II.12.2.

Pentru MT, corespunzător situației din fig. II.12.2.a, avem situația inițială reprezentată în fig. II.12.3.a. Configurațiile prin care trece succesiv mașina Turing sînt: $(s_{01}, (i_4, !)(i_5, i_3)(\mathbf{b}, i_2)(\mathbf{b}, i_1), 1) \vdash ((s_1, \downarrow), (i_6, !)(i_5, i_3)$ $(\mathbf{b}, i_2)(\mathbf{b}, i_1), 2) \vdash ((s_2, \downarrow), (i_6, !)(i_5, i_7)(\mathbf{b}, i_2)(\mathbf{b}, i_1), 3) \vdash (s_3, \downarrow), (i_6, !)$ $(i_5, i_7)(\mathbf{b}, i_2)(\mathbf{b}, i_1), 2) \vdash ((s_4, !), (i_6, !)(i_5, i_8)(\mathbf{b}, i_2)(\mathbf{b}, i_1), 1) \vdash ((s, \uparrow),$ $(i_9, !)(i_5, i_8)(\mathbf{b}, i_2)(\mathbf{b}, i_1), 2)$.

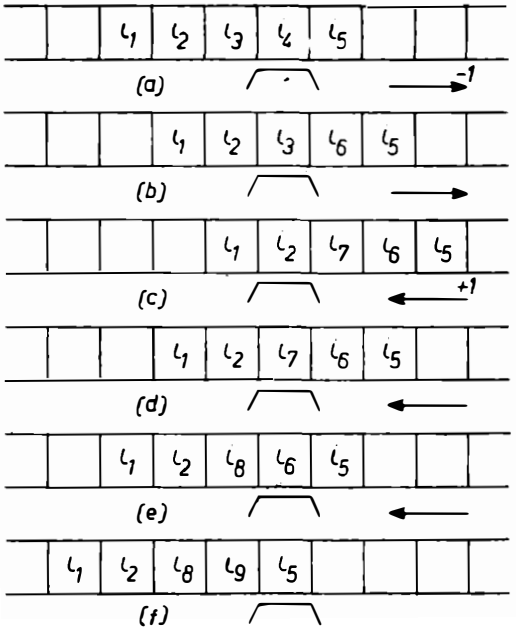

Fig. II. 12.2

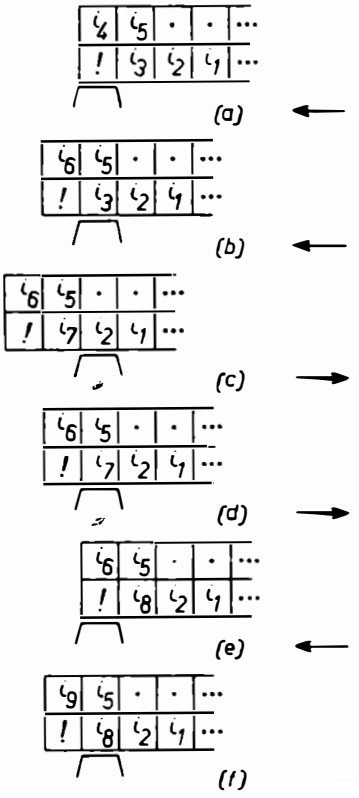

Fig. II. 12.3

Corespunzător, configurațiile succesive parcurse de mașina Turing unilaterală au fost reprezentate în figura II.12.3. Avem astfel $p \in L(MT)$, deci $L(MTBI) \subseteq L(MT)$.

Incluziunea inversă se poate arăta în același mod. ■

O altă variantă a mașinilor Turing se obține prin înlocuirea benzii unice și capului de citire-scriere unic cu o mulțime de benzi prevăzute fiecare cu cîte un cap de citire-scriere.

Difinița II.12.7. *O mașină Turing cu n benzi* este un 6-uplu $MTMMB = (n, I, S, f; \mathbf{b}, s_0)$, unde n este numărul de benzi, infinite în ambele sensuri, I este un alfabet finit, S o mulțime finită, nevidă, numită mulțimea stărilor, $f; (S \times \underbrace{I \times I \times \ldots \times I}_{n})' \to S \times \underbrace{I' \times \ldots I'}_{n} \times \{-1, 0, 1\}^n$ este funcția de trecere (cu $I' = I \cup \{\mathbf{b}\}$), $(S \times I \times \ldots \times I)' \subseteq S \times I \times \ldots \times I$, $\mathbf{b} \in I$ este simbolul „blanc", iar s_0 este starea inițială a mașinii.

Pentru o $MTMMB$ dispozitivul de comandă este prevăzut cu n capete de citire-scriere, cîte unul pentru fiecare bandă și diferitele benzi se pot mișca în diferite sensuri. Vom conveni că inițial prima bandă conține simbolii cuvîntului ce trebuie analizat, celelalte benzi conținînd doar simbolul blanc.

O configurație instantanee a unui $MTMMB$ este un $2n+1$-uplu $\gamma = (s, p_1, ..., p_n, m_1, ..., m_n)$, unde $s \in S$ este starea curentă a mașinii, $p_j \in (I \setminus \{\mathbf{b}\})^*$ este conținutul benzii j, iar m_j este poziția capului de la banda j socotită de la începutul cuvîntului p_j, $j = \overline{1, n}$.

Definiția II.12.8. *Limbajul acceptat de o* $MTMMB = (n, I, S, f; \mathbf{b}, s_0)$ *prin mulțimea de stări finale* $S_1 \subseteq S$ este:

$$L(MTMMB) = \{p \mid p \in \{I \setminus \{\mathbf{b}\}\}^*, \ (s_0, p, e, ..., e, k, 1, ..., 1) \overset{\bullet}{\vdash} (s, q_1, ... q_n, k_1, ..., k_n), s \in S_1\}.$$

Teorema II.12.2. *Un limbaj este acceptat de o* $MTMMB$ *dacă și numai dacă el este acceptat de o* MT.

Demonstrație. Deoarece orice MT poate fi concepută ca o $MTMMB$ cu $n = 1$, este limpede că limbajele acceptate de o MT sînt acceptate și de o $MTMMB$.

Vom arăta acum că un limbaj acceptat de o $MTMMB$ este acceptat de o MT. Să presupunem pentru fixarea ideilor că $n = 3$.

Vom construi o MT cu o bandă bilateral infinită avînd 6 piste (în general cu $2n$ piste) cîte două pentru fiecare bandă a mașinii Turing cu mai multe benzi, $MT = (I \cup \{1\}, S', f'; \mathbf{b}, s_0)$, unde $1 \notin I$, $S' = SI^{*3}$ și convenim să luăm $se = s$, $\forall s \in S$.

Vom folosi în continuare notațiile:

$$(\beta i) = \begin{cases} i, & \text{dacă } \beta = 1 \\ e, & \text{dacă } \beta = \mathbf{b} \end{cases}$$

$$\beta \dotplus \gamma = \begin{cases} 1 \dotplus \gamma, & \text{dacă } \beta = 1 \\ \gamma, & \text{dacă } \beta = \mathbf{b}. \end{cases}$$

$$\text{și } \beta\gamma \begin{cases} \gamma, & \text{dacă } \beta = 1. \\ 0, & \text{dacă } \beta = \mathbf{b}. \end{cases}$$

Dacă $f(s, i_1, i_2, i_3) = (\bar{s}, \bar{i}_1, \bar{i}_2, \bar{i}_3, \alpha_1, \alpha_2, \alpha_3)$, $\alpha_j \in \{-1, 0, 1\}$, $j = \overline{1, 3}$, vom defini $f'(s(\beta_1 i_1)(\beta_2 i_2)(\beta_3 i_3), (\gamma_1, i'_1, \gamma_2, i'_2, \gamma_3, i'_3))$ în tabelul II.12.5. Aici am considerat, la punctul 2, unul din cazurile posibile ($\alpha_1 = -1, \alpha_2 = \alpha_3 = 1$), celelalte tratîndu-se analog.

Anume, am considerat situația indicată în figura II.12.4 pentru $MTMMB$. Evoluția corespunzătoare pentru MT este descrisă în tabelul II.12.5.

Tabelul II.12.5

Nr. crt.	Situație	Valoarea lui $f(s(\beta_1 i_1)(\beta_2 i_2)(\beta_3 i_3), (\gamma, i'_1, \gamma, i'_2, \gamma, i'_3))$
1	$\beta_1 + \gamma_1 + \beta_2 + \gamma_2 + \beta_3 + \gamma_3 < 3$	$(s((\beta_2 + \gamma_1) i_1)((\beta_2 + \gamma_2) i_2)((\beta_3 + \gamma_3) i_3), (\gamma_1, i'_1, \gamma_2, i'_2, \gamma_3, i'_3), 1)$
2	$\beta_1 + \gamma_1 + \beta_2 + \gamma_2 + \beta_3 + \gamma_3 = 3$	
	a) $\max\{(\alpha_j \gamma_j) \mid j = \overline{1,3}\} = 1$ și, de exemplu $\alpha_2\gamma_2 = \alpha_3\gamma_3 = 1$	$(s(\beta_1 + \gamma_1 - \alpha_1\gamma_1) i_1)((\beta_2 + \gamma_2 - \alpha_2\gamma_2) i_2 (\beta_3 + \gamma_3 - \alpha_3\gamma_3) i_3), (\gamma_1, i_1, \mathbf{b}, \overline{i}_2, \mathbf{b}, \overline{i}_3), 1)$
3	$\beta_1 = 1, \gamma_1 = \gamma_2 = \gamma_3 = 0$	$(s(\beta_1 i_1), (0, i'_1, 1, i'_2, 1, i'_3), -1)$
4	$\gamma_1 \neq 1$	$(s(\beta_1 i_1), (0, i'_1, \gamma_2, i'_2, \gamma_3, i'_3), -1)$
5	$\gamma_1 = 1$	$(s, (0, \overline{i}_1, \gamma_2, i'_2, \gamma_3, i'_3), -1)$
6	$\gamma_1 = 0$	$(\overline{s}, (1, i'_1, \gamma_2, i'_2, \gamma_3, i'_3), 0)$

Să observăm (v. fig. II.12.5) că la un pas al $MTMMB$ corespund mai mulți pași executați de MT și anume : banda mașinii Turing MT se mișcă astfel încît prin dreptul capului de citire-scriere să treacă toată porțiunea în care apar simbolii 1 pe pistele de număr impar, starea schimbîndu-se succesiv : $s, si_1, si_1 i_2 i_3$ pentru a memora simbolii deasupra cărora se află 1. Capetele de citire-scriere ale mașinii Turing cu mai multe benzi se află tocmai în dreptul acestor simboli ; în pasul următor acești simboli sînt înlocuiți cu $\overline{i}_1, \overline{i}_2$ și $\overline{i}_3$ respectiv, iar capetele de citire-scriere își schimbă poziția. Noua situație a mașinii $MTMMB$ se va regăsi descrisă pe banda cu șase piste a mașinii MT. Într-adevăr, în continuare mașina MT schimbă ultimii simboli i_2 și i_3 în $\overline{i}_2, \overline{i}_3$, mută banda la stînga, înscrie pe pista a 3-a și a 5-a simbolul 1 (pentru a marca noile poziții ale capetelor 2 și 3) și apoi deplasează banda la dreapta.

Apoi, ajungînd în dreptul lui i_1 îl schimbă în $\overline{i}_1$, starea mașinii a redevenit s și simbolul 1 este șters. În sfîrșit MT mută banda la dreapta, înscrie pe prima pistă 1 și trece în starea s, noua stare a mașinii $MTMMB$. Astfel, dacă un cuvînt aflat pe prima bandă aduce $MTMMB$ într-o stare $s_1 \in S_1$, același cuvînt înscrie pe prima pistă a benzii MT (celelalte piste conținînd numai simbolul $\mathbf{b}$) aduce mașina MT tot în starea s_1.

Deci $L(MTMMB) \subseteq L(MT)$. Incluziunea inversă se arată analog. ■

Definiția II.12.9. O *mașină Turing nedeterministă* ($MTND$) este un 5-uplu

$$MTND = (I, S, f; \mathbf{b}, s_0)$$

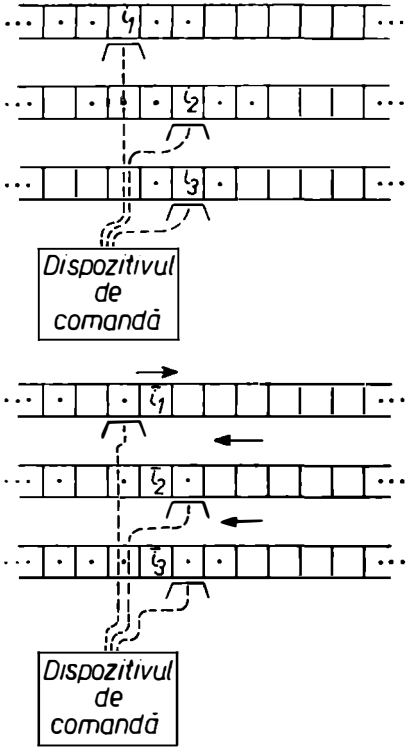

Fig. II. 12.4

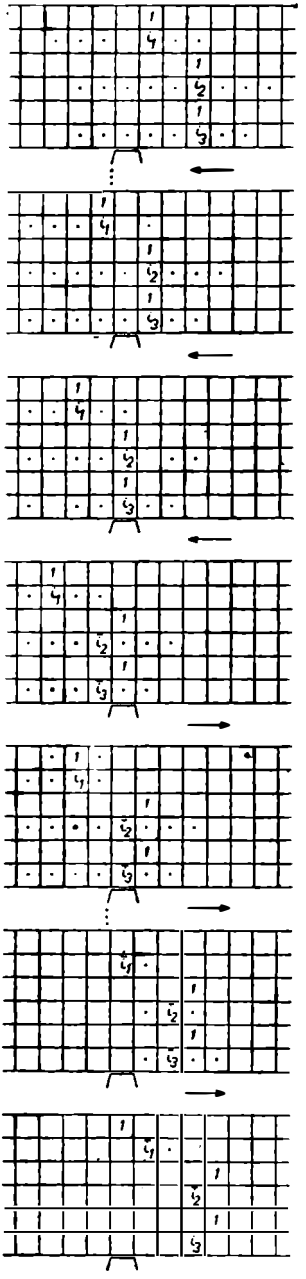

Fig. II. 12.5

în care I, S, $\mathbf{b}$, s_0 au aceleași semnificații ca în definiția II.12.1, iar $f: (S \times I)' \to \mathcal{P}^*(S \times (I \setminus \{\mathbf{b}\}) \times \{-1, 0, 1\})$ este funcția de trecere.

O $MTND$ are mai multe posibilități de alegere pentru noua stare în care trece, pentru simbolul care înlocuiește simbolul prelucrat și pentru deplasarea benzii. Noțiunea de configurație instantanee coincide cu cea introdusă pentru MT.

Definiția II.12.10. *Limbajul acceptat de o* $MTND = (I, S, f; \mathbf{b}, s_0)$ *prin mulțimea de stări finale* $S_1 \subseteq S$ este:

$$L(MTND) = \{p \mid p \in (I \setminus \{\mathbf{b}\})^*, \ \exists (s_0, p, k) \overset{*}{\vdash} (s_1, q, n), s_1 \in S_1\}.$$

Teorema II.12.3. *Un limbaj este acceptat de o* $MTND$ *dacă și numai dacă e acceptat de o* MT.

Demonstrație. Deoarece orice MT poate fi concepută ca o $MTND$, este clar că un limbaj acceptat de o MT este acceptat și de o $MTND$.

Vom arăta acum că un limbaj acceptat de o $MTND$ este acceptat de o MT.

Fie $p \in (I \setminus \{\mathbf{b}\})^*$ un cuvînt înscris pe banda $MTND$. La prelucrarea fiecărui simbol din p mașina $MTND$ are la dispoziție un număr finit de noi evoluții (schimbarea stării, înscrierea pe bandă și deplasarea benzii). Fie $r = \max\{|f(s, i)| \mid s \in S, i \in I\}$. La prelucrarea întregului cuvînt există, în total, cel mult $r^{l(p)}$ succesiuni posibile de configurații.

Să considerăm o mașină Turing cu 3 benzi:

$$MTMMB = (3, I \cup S \times I \times \{-1, 0, 1\}, S, f_1; \mathbf{b}, s_0).$$

Pe prima bandă a mașinii $MTMMB$ se înscriu simboli din I.

Banda a doua posedă trei piste destinate respectiv înscrierii simbolilor din mulțimile I, S și $\{-1, 0, 1\}$, iar cea de-a treia bandă va conține simboli din alfabetul I.

La depunerea unui cuvînt $p \in (I \setminus \{\mathbf{b}\})^*$ pe prima bandă $MTMMB$ „măsoară" lungimea cuvîntului p și apoi înscrie pe banda a doua toate evoluțiile succesive posibile ale mașinii $MTND$ care corespund cuvintelor de lungime $l(p)$, două șiruri de evoluții fiind separate prin blancuri.

În continuare, pentru fiecare șir de evoluții, se copie cuvîntul p pe banda a treia și, sub acțiunea simbolilor înscriși pe banda a doua, $MTMMB$ „simulează" pe banda a treia acțiunea $MTND$, înregistrînd și schimbările de stare comandate de tripletele înscrise pe banda a doua.

Cuvîntul p este acceptat de $MTND$ dacă și numai dacă există un șir de evoluții care conduce la o stare $s_1 \in S_1$, deci, dacă și numai dacă $MTMMB$ intră la un moment dat în starea s_1. Astfel, dacă p este acceptat de $MTND$, în mod necesar mașina cu mai multe benzi ajunge la un nou șir de evoluții înscris pe banda a doua care o conduce la o stare $s_1 \in S_1$ și reciproc. ■

Teorema II.12.4. *Un limbaj acceptat de o* MT *este de tip* 0; $L(MT) \in \mathcal{L}_0$.

Demonstrație. Fie $MT = (I, S, f; \mathbf{b}, s_0)$ o mașină Turing și $L(MT)$ limbajul acceptat de ea prin mulțimea de stări finale $S_1 \subseteq S$. Putem presupune

fără a restringe generalitatea că $\{S_1 \times \{I \setminus \{\mathbf{b}\}\}\} \cap (S \times I)' = \varnothing$, unde $(S \times I)'$ este domeniul de definiție al funcției f. (v. și observația II.12.3). Vom construi o gramatică de tip 0, care va genera limbajul $L(MT)$.

Fie $G = (I_N, I \setminus \{\mathbf{b}\}, x_0; F)$, unde $I_N = (\bar{I} \times I) \cup S \cup \{x_0, x_1, x_2\}$, $I = (I \setminus \{\mathbf{b}\}) \cup \{e\}$, iar F este dat de:

1) $x_0 \to s_0 x_1$;

2) $x_1 \to (i, i) x_1$, $\forall i \in I \setminus \{\mathbf{b}\}$;

3) $x_1 \to x_2$;

4) $x_2 \to (e, \mathbf{b}) x_2$;

5) $x_2 \to e$;

6) $s(i, i_1) \to (i, i_2) s'$ pentru orice $i \in \bar{I}$, $s \in S$, $i_1 \in I$, astfel încît $f(s, i_1) = (s', i_2, 1)$;

7) $(i_1, i_2) s (i_3, i_4) \to s' (i_1, i_2) (i_3, i_5)$ pentru orice $i_4, i_5, i_2 \in I$, $i_1, i_3 \in \bar{I}$, $s \in S$, astfel încît $f(s, i_4) = (s', i_5, -1)$;

8) $(i_1, i_2) s \to s i_1 s$, $s (i_1, i_2) \to s i_1 s$ și $s \to e$ pentru $\forall i_1 \in \bar{I}$, $i_2 \in I$ și $s \in S_1$.

Să arătăm că $L(MT) \subseteq L(G)$. Fie $p = i_1 i_2 \dots i_n \in L(MT)$. Să presupunem că în procesul recunoașterii cuvîntului p, MT folosește un număr de m celule situate la dreapta celulei ocupată inițial de simbolul i_n. Atunci, utilizînd regulile de tipul 1) și 2) din gramatica G, obținem $x_0 \overset{*}{\Rightarrow} s_0 (i_1, i_1) (i_2, i_2) \dots (i_n, i_n) x_1$, unde $i_j \in I \setminus \{\mathbf{b}\}$, $j = \overline{1, n}$. Apoi, folosind regula 3), regula 4) de m ori și, în fine, regula 5), obținem:

$$x_0 \overset{*}{\Rightarrow} s_0 (i_1, i_1) (i_2, i_2) \dots (i_n, i_n) (e, \mathbf{b})^m.$$

În continuare putem folosi doar regulile de forma 6) și 7), care simulează activitatea mașinii Turing pînă ce se ajunge la generarea unui cuvînt care conține simbolul unei stări finale, din mulțimea S_1. Să observăm că acest proces nu afectează primele componente ale perechilor (i_j, i_j). Vom arăta, prin inducție, după numărul t de tacte în care acționează mașina MT că, dacă $(s_0, i_1 i_2 \dots i_n, 1) \overset{*}{\vdash} (s, i'_1, \dots i'_k, h)$, atunci:

(II.12.1) $s_0 (i_1, i_1) (i_2, i_2) \dots (i_n, i_n) (e, \mathbf{b})^m \overset{*}{\Rightarrow} (i_1, i'_1) (i_2, i'_2) \dots (i_{k-1}, i'_{k-1}) s_1 (i_k, i'_k) \dots (i_{n+m}, i'_{n+m})$

cu $i_1, \dots, i_n \in I \setminus \{\mathbf{b}\}$, $i_{n+1} = \dots i_{n+m} = e$, $i'_1, \dots, i'_{n+m} \in I$ și $i'_{k+1} = \dots i'_{n+m} = \mathbf{b}$.

Afirmația se verifică în mod evident pentru $t = 0$. S-o presupunem adevărată pentru $t - 1$ tacte și fie $(s_0, i_1 \dots i_n, 1) \overset{*}{\vdash} (s_1, i'_1 \dots i'_h, j_1) \vdash (s_2, i''_1 \dots i''_h, j_2)$ o evoluție a mașinii Turing executată în t tacte.

Prin ipoteza inductivă avem $s_0 (i_1, i_1) \dots (i_n, i_n) (e, \mathbf{b})^m \overset{*}{\Rightarrow} (i_1, i'_1) \dots$

$\ldots s_1 (i_{j_1}, i'_{j_1}) \ldots (i_{n+m}, i'_{n+m})$. Punînd $j_2 = j_1 + \alpha$ (unde $\alpha \in \{-1, 1\}$) avem $f(s_1, i'_{j_1}) = (s_2, i''_{j_1}, \alpha)$, de unde folosind corespunzător 6) sau 7), după cum $\alpha = 1$ sau $\alpha = -1$, obținem $s_1 (i_{j_1}, i'_{j_1}) \to (i'_{j_1}, i''_{j_1}) s_2$ sau $(i_{j_1-1}, i'_{j_1-1}) s_1 (i_{j_1}, i'_{j_1}) \to s_2 (i_{j_1-1}, i'_{j_1-1}) (i_{j_1}, i''_{j_1})$ și

$i'_j = i''_j$, $\forall j, j \neq j_1$. Astfel $s_0 (i_1, i_1) \ldots (i_n, i_n)$ $(e, \mathbf{b})^m \overset{*}{\Rightarrow} (i_1, i''_1) \ldots$ $\ldots s_2 (i_{j_2}, i''_{j_2}) \ldots (i_{n+m}, i''_{n+m})$, verificîndu-se afirmația inițială.

Dacă $s_1 \in S_1$ aplicînd reguli de generare de tipul 8) în (II.12.1) obținem:

$s_0 (i_1, i_1) (i_2, i_2) \ldots (i_n, i_n) (e, \mathbf{b})^m \overset{*}{\Rightarrow} (i_1, i'_1) (i_2, i'_2) \ldots (i_{k-1}, i'_{k-1}) s_1$ $(i_k, i'_k) \ldots (i_{n+m}, i'_{n+m}) \Rightarrow (i_1, i'_1) (i_2, i'_2) \ldots (i_{k-1}, i'_{k-1}) s_1 i_k s_1 (i_{k+1},$ $i'_{k+1}) \ldots (i_{n+m}, i'_{n+m}) \Rightarrow (i_1, i'_1) (i_2, i'_2) \ldots (i_{k-2}, i'_{k-2}) s_1 i_{k-1} s_1 s_1 i_k s_1 s_1 i_{k+1}$

$s_1 (i_{k+2}, i'_{k+2}) \ldots (i_{n+m}, i'_{n+m}) \overset{*}{\Rightarrow} i_1 \ldots i_n$. Astfel $p = i_1 i_2 \ldots i_n \in L(G)$.

Invers, fie $p \in L(G)$. Vom arăta că $p \in L(MT)$.

Să observăm că pentru ca $p \in L(G)$ în mod necesar deducția $x_0 \overset{*}{\Rightarrow} p$ se împarte în trei: $x_0 \overset{*}{\Rightarrow} r \overset{*}{\Rightarrow} q \overset{*}{\Rightarrow} p$, unde în deducția $x_0 \overset{*}{\Rightarrow} r$ se aplică exclusiv regulile de generare de tipurile 1)–5), în $r \overset{*}{\Rightarrow} q$ se folosesc numai reguli de generare de tipurile 6)–7), iar în deducția $q \overset{*}{\Rightarrow} p$ utilizăm numai regulile de generare de tipul 8). Cum regulile 6)–7) nu schimbă prima componentă a oricărei perechi (i_j, i'_j), rezultă că r are în mod necesar forma:

$r = s_0 (i_1, i_1) \ldots (i_n, i_n) (e, \mathbf{b})^m$ unde m poate fi ales arbitrar, convenabil pentru partea a doua a deducției. Cuvîntul q va avea forma $q = (i_1, i'_1) \ldots$ $\ldots s (i_k, i'_k) \ldots (i_{n+m}, i'_{n+m})$ cu $i_{n+1} = \ldots i_{n+m} = e$, $i'_1 \ldots, i'_{n+m} \in I$ și $i'_{k+1} = \ldots$ $\ldots = i'_{n+m} = \mathbf{b}$, iar $s \in S_1$. Să arătăm, prin inducție, după numărul t de aplicări ale regulilor de generare de tip 6) sau 7) că dacă există deducția:

$$s_0 (i_1, i_1) \ldots (i_n, i_n) (e, \mathbf{b})^m \overset{*}{\Rightarrow} (i_1, i''_1) \ldots (i_{k-1}, i''_{k-1}) s (i_k, i''_k) \ldots (i_{n+m}, i''_{n+m}),$$

atunci avem în MT calculul $(s_0, i_1 \ldots i_n, 1) \overset{*}{\vdash} (s, i''_1 \ldots i''_k, k)$. Pentru $t = 0$ afirmația este banal adevărată.

Să presupunem deci afirmația adevărată pentru $t - 1$ și fie deducția $s_0 (i_1, i_1) \ldots (i_n, i_n) (e, \mathbf{b})^m \overset{*}{\Rightarrow} (i_1, i''_1) \ldots s'' (i_{k-1}, \hat{i}_{k-1}) (i_k, i''_k) \ldots$ $\ldots (i_{n+m}, i''_{n+m}) \Rightarrow (i_1, i''_1) \ldots (i_{k-1}, i''_{k-1}) s (i_k, i''_k) \ldots$. $\ldots (i_{n+m}, i''_{n+m})$, în care s-au aplicat t reguli de generare, ultima regulă de generare aplicată fiind regula de tipul 6): $s'' (i_{k-1}, \hat{i}_{k-1}) \to (i_{k-1}, i''_{k-1}) s$. În baza ipotezei inductive există calculul $(s_0, i_1 \ldots i_n, 1) \overset{*}{\vdash} (s'', i''_1 \ldots \hat{i}''_{k-2} i_{k-1}, k-1)$ și deoarece $f(s'', i_{k-1}) = (s, i''_k, 1)$ găsim $(s_0, i_1 \ldots i_n, 1) \overset{*}{\vdash} (s, i''_1 \ldots$ $\ldots i''_{k-1}, k)$.

Similar se procedează dacă ultima regulă de generare aplicată ar fi de tipul 7). Prin urmare, deducției $r \overset{*}{\Rightarrow} q$ îi corespunde în mașina Turing cal-

culul $(s_0, r, 1) \overset{\bullet}{\vdash} (s, q, h)$. Cum aplicarea regulilor 8) nu se poate face decît dacă $s \in S_1$, rezultă că $p \in L(MT)$. ■

§ II. 13. MAŞINI TURING—DISPOZITIVE DE CALCUL

Un alt rol jucat de maşinile Turing este acela de a „calcula" valorile unor funcţii finite pe semigrupuri libere cu valori în semigrupuri libere.

Fie $MT = (I, S, f; \mathbf{b}, s_0)$ o maşină Turing. Funcţionarea ei, descrisă de funcţia f, se poate reprezenta şi prin indicarea unei mulţimi finite Q de şiruri care conţin cinci simboluri.

Astfel, dacă $(s, i) \in (S \times I)'$ şi $f(s, i) = (s', i', \alpha)$, avem $sis'i' \in Q$.

Dacă două şiruri $sis'i'\alpha$ şi $s_1 i_1 s_1' i_1' \alpha_1$ au $s = s_1$ şi $i = i_1$, atunci $s' = s_1'$, $i' = i_1'$, $\alpha = \alpha_1$, deoarece maşina Turing funcţionează în mod determinist şi perechea (s, i) determină în mod unic tripletul $(s', i', \alpha) = f(s, i)$. Deoarece mulţimea Q şi funcţia f se determină reciproc în mod unic putem da maşina Turing şi sub forma $MT = (I, S, Q; \mathbf{b}, s_0)$. Mulţimea Q va mai fi numită *programul maşinii Turing*.

Vom nota configuraţiile instantanee (s, p, n) şi prin qsr, unde $l(q) = n - 1$ şi $p = qr$, punînd astfel în evidenţă poziţia capului de citire-scriere, el găsindu-se în dreptul celulei ocupate de primul simbol din cuvîntul r.

O configuraţie instantanee $qsr = qsir_1$ (sau qs) este *configuraţie instantanee terminală*, dacă nu există vreun 5-uplu din Q, avînd pe primele două locuri si (respectiv $s\mathbf{b}$). Starea s este în acest caz o *stare terminală*.

În cele ce urmează, pînă la afirmaţia contrară, vom presupune că toate maşinile Turing considerate au o singură *stare terminală*.

Definiţia II.13.1. *Funcţia calculată pe I^* de maşina $MT = (I, S, Q; \mathbf{b}, s_0)$* este aplicaţia $F: (I^*)' \to I$ dată prin

$$F(p) = \begin{cases} qr, \text{ dacă } s_0 p \overset{\bullet}{\vdash} qsr, s \in S \text{ şi } qsr \text{ este o configuraţie instantanee terminală} \\ \text{nedefinită, dacă nu există un astfel de calcul,} \end{cases}$$

unde $(I^*)' \subseteq I$.

Vom introduce o nouă funcţie prin eliminarea simbolului $\mathbf{b}$, care nu este un purtător propriu-zis de informaţie.

Definiţia II.13.2. *Funcţia calculată* pe $[(I \setminus \{\mathbf{b}\})^*]^n$ *de maşina* $MT = (I, S, Q; \mathbf{b}, s_0)$ este aplicaţia $F^n: ([(I \setminus \{\mathbf{b}\})^*]^n)' \to (I \setminus \{\mathbf{b}\})^*$ dată prin

$$F^n(p_1, \ldots, p_n) = \begin{cases} q, \text{ dacă } F(p_1 \mathbf{b} p_2 \mathbf{b} \ldots \mathbf{b} p_n) = uqv \text{ cu} \\ u, v \in \{\mathbf{b}\}^*, \\ \text{nedefinită, în caz contrar,} \end{cases}$$

unde, $([(I \setminus \{\mathbf{b}\})^*]^n)' \subseteq [(I \setminus \{\mathbf{b}\})^*]^n$.

Definiția II.13.3. O *funcție parțial calculabilă de o mașină Turing* este o aplicație $h : ([(I \setminus \{\mathbf{b}\})^*]^n)' \to (I \setminus \{\mathbf{b}\})^*$ pentru care există o mașină Turing a cărei funcție calculată F^n coincide cu h, pe domeniul de definiție al lui h.

Aplicația h este *total calculabilă*, dacă este total definită.

Rezultatul central al acestui paragraf este faptul că orice funcție parțial recursivă pe un alfabet I este calculabilă de o mașină Turing. În acest scop este necesară în prealabil considerarea unor mașini Turing simple care calculează anumite funcții parțial recursive speciale pe un alfabet I și introducerea operației de compunere a mașinilor Turing.

Fie $MT_1 = (I, \{s_{01}, ..., s_{n-2}, s\}, Q_1; \mathbf{b}, s_{01})$ și $MT_2 = (I, \{s_{02}, s_1', ..., s_{m-2}', s\}, Q_2; \mathbf{b}, s_{02})$ două mașini Turing cu același alfabet de intrare I și aceeași stare terminală s.

Definiția II.13.4. *Mașina Turing* obținută prin *compunerea mașinilor Turing* MT_1 și MT_2 este mașina $MT = MT_1 * MT_2 = (I, S, Q; \mathbf{b}, s_{01})$, unde $S = S_1 \cup \bar{S}_2$ cu $\bar{S}_2 = \{s_{n+1}, ..., s_{n+m-1}\}$, iar programul Q constă din:

i) șirurile lui Q_1 în care starea terminală s este înlocuită cu starea s_{n+1};

ii) șirurile din Q_2 în care s_{02} este înlocuit prin s_{n+1}, iar s_j' prin s_{n+j+1} starea terminală s rămînînd neschimbată.

Teorema II.13.1. *Fie* $p_j, q_j \in I^*$, $j = \overline{1, 3}$ și $p_1 s_{01} q_1 \overset{*}{\underset{MT_1}{\vdash}} p_2 s q_2$, $p_2 s_{02} q_2 \overset{*}{\underset{MT_2}{\vdash}} p_3 s q_3$. *Atunci* $p_1 s_{01} q_1 \overset{*}{\underset{MT}{\vdash}} p_3 s q_3$, *unde* $MT = MT_1 * MT_2$.

Demonstrație. În baza definiției II.13.4, în mașina MT avem calculul $p_1 s_{01} q_1 \overset{*}{\underset{MT}{\vdash}} p_2 s_{n+1} q_2$. Apoi schimbînd șirurile din Q_2 care participă la calculul $p_2 s_{02} q_2 \overset{*}{\underset{MT_2}{\vdash}} p_3 s q_3$, conform aceleiași definiții, obținem $p_2 s_{n+1} q_2 \overset{*}{\underset{MT_1}{\vdash}} p_3 s q_3$, de unde $p_1 s_{01} q_1 \overset{*}{\underset{MT}{\vdash}} p_3 s q_3$.

Prezentăm în continuare unele mașini Turing simple care calculează anumite funcții parțial recursive speciale pe un alfabet I. Pentru simplitatea expunerii vom lua $I = \{\mathbf{b}, i\}$, cazul general tratîndu-se analog.

Exemplul II.13.1. Mașina Turing care „transportă" blancul este mașina $T = (I, \{s_0, s_1, s_2, s_3, s_4, s_5, s\}, Q; \mathbf{b}, s_0)$, unde $Q = \{s_0 \mathbf{b} s_1 \mathbf{b} 1, s_1 \mathbf{b} s_2 i 0, s_2 i s_2 i 1, s_2 \mathbf{b} s_3 \mathbf{b} - 1, s_3 i s_4 \mathbf{b} 0, s_4 \mathbf{b} s_5 \mathbf{b} - 1, s_5 i s_5 i - 1, s_5 \mathbf{b} s \mathbf{b} 0\}$.

Avem $s_0 \mathbf{b} b i^n \mathbf{b} \overset{*}{\underset{T}{\vdash}} s \mathbf{b} 1^n \mathbf{b} b$. Într-adevăr, avem succesiv $s_0 \mathbf{b} b i^n \mathbf{b} \overset{*}{\vdash} \mathbf{b} s_1 \mathbf{b} i^n \mathbf{b} \overset{*}{\vdash} \mathbf{b} i i^n s_2 \mathbf{b} \overset{*}{\vdash} \mathbf{b} i^n s_4 \mathbf{b} \mathbf{b} \overset{*}{\vdash} s_5 \mathbf{b} i^n \mathbf{b} \mathbf{b} \overset{*}{\vdash} s \mathbf{b} i^n \mathbf{b} \mathbf{b}$.

Exemplul II.13.2. Mașina Turing care deplasează la dreapta conținutul benzii este mașina $D^- = (I, \{s_0, s_1, s\}, Q; \mathbf{b}, s_0)$ unde $Q = \{s_0 \mathbf{b} s_1 \mathbf{b} - 1, s_1 i s_1 i - 1, s_1 \mathbf{b} s \mathbf{b} 0\}$. Această mașină efectuează transformarea $\mathbf{b} i^n s_0 \mathbf{b} \overset{*}{\underset{D}{\vdash}} s \mathbf{b} i^n \mathbf{b}$, deoarece avem $\mathbf{b} i^n s_0 \mathbf{b} \vdash \mathbf{b} i^{n-1} s_1 i \mathbf{b} \overset{*}{\vdash} \mathbf{b} s_1 i^n \mathbf{b} \vdash s_1 \mathbf{b} i^n \mathbf{b} \vdash s \mathbf{b} i^n \mathbf{b}$.

Exemplul II.13.3. Mașina Turing care deplasează la stînga conținutul benzii este mașina $D^+ = (I, \{s_0, s_1, s\}, Q; \mathbf{b}, s_0)$, unde $Q = \{s_0 \mathbf{b} s_1 \mathbf{b} 1,$

$s_1 is_1 i\,1, s_1\,bsb0\}$. Avem $s_0\ bi^n\ b \overset{*}{\underset{D^+}{\vdash}} bi^n\ sb$, deoarece $s_0\ bi^n\ b \vdash bs_1\ i^n\ b \overset{*}{\vdash} bi^n\ s_1\ b \vdash bi^n\ sb$.

Exemplul II.13.4. Mașina Turing care permută două cuvinte de pe bandă este mașina $P = (I, \{s_0, s_1, ..., s_{18}, s\}, Q; b, s_0)$, în care Q va fi descris în continuare. Această mașină realizează calculul $bi^n\ s_0\ bi^m\ b \overset{*}{\vdash} bi^m\ sbi^n\ b$.

Vom nota cu $Q(MT)$ șirurile din mulțimea Q a mașinii MT. Fie $m > 0$. Vom include în $Q(P)$ șirurile din $Q(D^+)$, în care vom schimba pe s cu s_2. Obținem astfel calculul $bi^n\ s_0\ bi^m\ b \overset{*}{\underset{P}{\vdash}} bi^n\ bi^m\ s_2 b$.

Mai departe, considerăm șirurile $s_2\ bs_3\ b - 1$, $s_3 is_4\ b0$, $s_4\ bs_5\ b - 1$, $s_5 is_5\ i - 1$, $s_5\ bs_6\ i0$, în urma aplicării cărora putem continua calculul, obținînd: $bi^n bi^m s_2\ b \overset{*}{\vdash} bi^n s_5\ bi^{m-1} bb \vdash bi^n s_6\ i^m bb$.

Pentru a obține configurația $bi^n s_6\ i^m bb$ și pentru $m = 0$, adăugăm șirul $s_3\ bs_6\ b0$.

În continuare folosim șirurile $s_6 is_7\ i - 1$, $s_7 is_8\ b0$, $s_8\ bs_9\ b1$, $s_9 is_9\ i1$, $s_9\ bs_{10}\ i0$, $s_{10}\ is_{11}\ i - 1$, $s_{11}\ is_{12}\ b0$, $s_{12}\ bs_{13}\ b - 1$, $s_{13}\ is_{13}\ i - 1, s_{13}\ bs_{14}\ i0$, în urma cărora avem calculul: $bi^n\ s_6\ i^m bb \vdash bi^{n-1} s_7\ ii^m\ bb \overset{*}{\vdash} bi^{n-1}\ bi^m\ s_9\ bb \overset{*}{\vdash} bi^{n-1}\ bi^{m-1}\ s_{12}\ bib \overset{*}{\vdash} bi^{n-1} s_{13}\ bi^{m-1}\ bib \vdash bi^{n-1} s_{14}\ i^m bib$.

Am presupus aici $n > 0$, $m > 0$. Pentru a cuprinde și cazul $m = 0$, adăugăm șirurile $s_6\ bs_{15}\ i0$, $s_{15}\ is_{16}\ i - 1$, $s_{16}\ is_{14}\ b0$.

Cu ajutorul șirurilor $s_{14}\ is_6\ i0$, $s_{14}\ bs_6\ b0$, obținem configurația $bi^{n-1} s_6\ i^m\ bib$. Dacă $n > 1$, atunci această configurație se transformă în $bi^{n-2} s_6\ i^m\ biib$. Dacă $n > 2$, repetînd procesul, obținem $bi^{n-3} s_6\ i^m\ biiib$, ș.a.m.d., pînă ce obținem, după n aplicări ale procedeului descris, configurația $bs_6\ i^m\ bi^n\ b$. Transformarea acestei configurații se face cu ajutorul șirului $s_6 is_7\ i - 1$ (dacă $m > 0$), sau $s_6\ bs_{15}\ i0$ (dacă $m = 0$), obținînd configurațiile $s_7\ bi^m\ bi^n$, sau, respectiv, $s_{16}\ b\ ii^n\ b$.

În sfîrșit, șirurile $s_7\ bs_{17}\ b1$, $s_{17}\ is_{17}\ i1$, $s_{17}\ bsb0$ și $s_{16}\ bs_{18}\ b1$, $s_{18}\ isb0$ conduc respectiv la $bi^m sbi^n b$, după cum $m > 0$, sau $m = 0$.

Exemplul II.13.5. Pentru dedublarea unui cuvînt de pe bandă vom folosi mașina Turing $M = (I, S, Q; b, s_0)$, unde S și Q vor fi precizați ulterior. Această mașină realizează calculul $s_0\ bi^n\ b^{n+3} \overset{*}{\underset{M}{\vdash}} sbi^n\ bi^n\ bb$.

Șirurile $s_0\ b\ s_1\ b1$, $s_1 is_1\ i1$ au ca efect calculul $s_0\ bi^n\ b^{n+3} \overset{*}{\vdash} bi^n\ s_1\ b^{n+3} = bi^n\ s_1\ bb^0\ bb^0\ b^{n+1}$, unde $b^0 = e$.

Mulțimea de șiruri $s_1\ bs_2\ b - 1$, $s_2 is_3\ b0$, $s_3\ bs_4\ b1$, $s_4\ bs_5\ i0$, $s_5 is_5\ i1$, $s_5 bs_6\ b1$, $s_6 is_6\ i1$, $s_6\ bs_7\ i0$, $s_7 is_7\ i - 1$, $s_7\ bs_8\ b - 1$, $s_8 is_8\ i - 1$, transformă configurația $bi^{n-h}\ s_1\ bi^h\ bi^h\ b$ în $bi^{n-(h+1)}\ s_8\ bi^{h+1}\ b\ i^{h+1}$ (exercițiu!), în cazul în care $n - h > 0$.

Adăugînd șirul $s_8\ bs_1\ b0$ se parcurge din nou mulțimea de șiruri de mai sus, obținîndu-se $bi^{n-(h+2)}\ s_1\ bi^{h+2}\ bi^{h+2}$ ș.a.m.d., pînă cînd, după n repetări ale procedeului, se ajunge la configurația $bs_1\ bi^n\ bi^n\ b$.

Şirul $s_1\,\mathbf{b}s_2\,\mathbf{b}-1$ conduce maşina la configuraţia $s_2\,\mathbf{b}\mathbf{b}i^n\,\mathbf{b}i^n\,\mathbf{b}$. Considerînd acum şirurile obţinute din $Q(T)$ prin înlocuirea stărilor $\{s_0, s_1, s_2, s_3, s_4, s_5, s\}$, respectiv cu $\{s_2, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}\}$, ajungem la configuraţia $s_{14}\,\mathbf{b}i^n\,\mathbf{b}\mathbf{b}i^n\,\mathbf{b}$.

Folosim acum şirurile din $Q(D^+)$, în care înlocuim s_0, s_1, s prin s_{14}, s_{15}, s_{16}, respectiv. Ajungem la configuraţia $\mathbf{b}i^n s_{16}\,\mathbf{b}\mathbf{b}i^n\,\mathbf{b}$. Aplicînd din nou şirurile din $Q(T)$, în care stările $s_0, s_1, \ldots, s_5, s$ le înlocuim respectiv prin $s_{16}, s_{17}, \ldots, s_{21}, s_{22}$, obţinem $\mathbf{b}i^n s_{22}\,\mathbf{b}i^n\,\mathbf{b}\mathbf{b}$. În sfîrşit, utilizînd şirurile din $Q(D^-)$, în care stările s_0, s_1, s sînt schimbate respectiv prin s_{22}, s_{23}, s obţinem $s\mathbf{b}i^n\,\mathbf{b}i^n\,\mathbf{b}\mathbf{b}$.

Observaţia II.13.1. Maşinile Turing considerate în exemplele precedente transformă conţinutul benzii fără a folosi celule situate la stînga primei celule a conţinutului iniţial. Acest mod de funcţionare este cunoscut în literatura de specialitate sub numele de *calcul corect.*

Observaţia II.13.2. Compunînd maşinile D^-, D^+, P, M obţinem alte maşini Turing. Anume, maşina $D^+ * D^+$ realizează calculul $s_0\,\mathbf{b}i^n\,\mathbf{b}i^m\,\mathbf{b} \vdash \mathbf{b}i^n\,\mathbf{b}i^m\,s\mathbf{b}$. Apoi, maşina $C = P * D^- * P$ execută transformarea $\mathbf{b}i^n\,\mathbf{b}i^m\,s_0\,\mathbf{b}i^h\,\mathbf{b} \overset{*}{\vdash} \mathbf{b}i^h\,s\mathbf{b}i^n\,\mathbf{b}i^m\,\mathbf{b}$. În sfîrşit, $K = D^+ * M * P * D^+ * P' * M * P * D^+ * P * D^- * D^- * P * D^+$ efectuează calculul $s_0\,\mathbf{b}i^n\,\mathbf{b}i^m \overset{*}{\vdash} \mathbf{b}i^n\,\mathbf{b}i^m\,s\mathbf{b}i^n\,\mathbf{b}i^m$, aşa cum se poate constata imediat.

Observaţia II.13.3. Pentru simplificarea scrierii calculelor care intră în demonstraţii, vom face următoarea convenţie de notaţie. Fie $MT = (I, S, Q; \mathbf{b}, s_0)$ o maşină Turing cu $S = \{s_0, s_1, \ldots, s_{n-1}, s\}$. Vom nota cu Q_k mulţimea şirurilor care se obţin din şirurile lui Q, înlocuind simbolii s_j cu simbolii s_{j+k}, $j = \overline{0, n-1}$, iar s prin s_{n+k}, iar cu $M(T)_k$ maşina $(I, \{s_k, \ldots, s_{n+k-1}, s_{n+k}\}, Q_k; b, s_k)$.

Dacă $ps_0\,q \underset{MT}{\overset{*}{\vdash}} rsu$ avem $ps_k\,q \underset{(MT)_k}{\overset{*}{\vdash}} rs_{n+k}\,u$, ceea ce notăm prin $ps_kq\,(MT)_k\,rs_{n+k}\,u$. Corespunzător acestei notaţii, pentru compunerea de maşini avem: dacă $p_0\,s_0\,q_0\,(U)_0\,p_1\,s_n\,q_1\,(V)_n\,p_2\,s_m\,q_2 \ldots (W)_l\,p_j\,s_n\,q_j$ atunci $p_0\,s_0\,q_0\,((U * V) \ldots * W)\,p_j\,sq$.

Teorema II.13.2. *Funcţiile primare* $\mathbf{o}$, $\mathbf{s}_i$ şi $\mathbf{i}_m^k$ *sînt calculabile de maşini Turing.*

Demonstraţie. O maşină Turing $MT_{\mathbf{o}} = (\{\mathbf{b}, i\}, S, Q; \mathbf{b}, s_0)$ calculează funcţia $\mathbf{o}$ dacă ea poate realiza calculul $s_0\,\mathbf{b}i^n\,\mathbf{b} \overset{*}{\vdash} s\mathbf{b}\mathbf{b}^{n+1}$, ceea ce impune prezenţa în Q a următoarelor şiruri $s_0\,\mathbf{b}s_1\,\mathbf{b}1$, $s_1 i s_1 i 1$, $s_1\,\mathbf{b}s_2\,\mathbf{b}-1$, $s_2 i s_3\,\mathbf{b}0$, $s_3\,\mathbf{b}s_2\,\mathbf{b}-1$, $s_2\,\mathbf{b}s\mathbf{b}0$. Într-adevăr, în acest caz avem:

$$s_0\,\mathbf{b}i^n\,\mathbf{b} \overset{*}{\vdash} \mathbf{b}i^n\,s_1\,\mathbf{b} \overset{*}{\vdash} \mathbf{b}i^{n-1}\,s_3\,\mathbf{b}\mathbf{b} \overset{*}{\vdash} s_2\,\mathbf{b}\mathbf{b}^{n+1} \vdash s\mathbf{b}\mathbf{b}^{n+1}.$$

Apoi maşina Turing $MT_{\mathbf{s}} = (\{\mathbf{b}, i\}, S, Q; b, s_0)$ calculează $s_0\,\mathbf{b}i^n\,\mathbf{b} \overset{*}{\vdash} s\mathbf{b}i^{n+1}$, dacă Q conţine următoarele şiruri:
$s_0\,\mathbf{b}s_1\,\mathbf{b}1$, $s_1\,\mathbf{b}s_1\,\mathbf{b}1$, $s_1\,\mathbf{b}s_2\,i0$, $s_2 i s_2 i - 1$, $s_2\,\mathbf{b}s\mathbf{b}0$, deoarece avem $s_0, \mathbf{b}i^n\,\mathbf{b} \overset{*}{\vdash} \overset{*}{\vdash} \mathbf{b}i^n\,s_1\,\mathbf{b} \overset{*}{\vdash} s\mathbf{b}i^{n+1}$.

Ne vom mărgini, pentru claritate, la cazul funcției i_2^2. Avem $s_0\ bi^n\ bi^m\ (D^+)_0\ bi^n\ s_2\ bi^m\ (P)_2\ bi^m\ s_{22}\ bi^n\ (MT_0)_{22}\ bi^m\ s_{26}\ b...\ b\ b\ (D^-)_{26}\ s_{28}\ bi^m$. Conchidem că funcția i_2^2 este calculată de mașina $D^+ * P * MT_0 * D^-$.

Teorema II.13.3. *Dacă funcțiile parțiale* $g(x_1, ..., x_n)$, $f_1(x_1, ..., x_k), ..., f_n(x_1, ..., x_k)$ *sînt calculabile de mașini Turing, atunci funcția obținută prin superpoziție* $f(x_1, ..., x_k) = g(f_1(x_1, ... x_k), ..., f_n(x_1, ..., x_k))$ *este calculabilă de o mașină Turing.*

Demonstrație. Vom considera cazul $n = k = 2$, cazul general tratîndu-se analog. Să presupunem că funcțiile g, f_1, f_2 sînt calculate respectiv de mașinile Turing $MT(g)$, $MT(f_1)$, $MT(f_2)$. Avem :

$$s_0\ bi^n\ bi^m\ (k)_0\ bi^n\ bi^m\ s_\alpha\ bi^n\ bi^m\ (MT(f_1))_\alpha\ bi^n\ bi^m\ s_\beta\ bi^{f_1(n,m)}\ (C)_\beta\ bi^{f_1(n,m)}\ s_\gamma\ bi^n$$

$$bi^m\ (MT(f_2))_\gamma\ bi^{f_1(n,m)}\ s_\delta\ bi^{f_1(n,m)}\ (D^-)_\delta\ s_\varepsilon\ bi^{f_1(n,m)}\ bi^{f_1(n,m)}\ (MT(g))_\varepsilon$$

$$s_\tau\ bi^{g(f_1(n,m),\ f_2(n,m))}.$$

Adăugînd șirul $s_\tau\ bsb0$, obținem configurația finală :

$$sbi^{g(f_1, f_2)}. \blacksquare$$

Teorema II.13.4. *Dacă funcțiile* g *și* h *sînt calculabile de mașini Turing, atunci funcția* $f = R(g, h)$, *obținută prin recursie primitivă, este calculabilă de o mașină Turing.*

Demonstrație. Ne vom mărgini la considerarea recursiei primitive aplicată asupra funcțiilor $g(x)$, de o variabilă, și $h(x, i, z)$ de trei variabile, care ne furnizează funcția de două variabile f definită prin : $f(x, 0) = g(x)$, $f(x, i + 1) = h(x, i, f(x, i))$.

Notăm $h(x, i, f(x, i)) = k(f(x, i), x)$. Fie $MT(g)$ și $MT(k)$ mașinile Turing care calculează funcțiile g și k, respectiv. Avem $s_0\ bi^n\ bi^m\ b\ (D^+ * P * M * P * D^+)_0\ bi^n\ bi^m\ s_\alpha\ bi^n\ b\ (MT(g))_\alpha\ bi^n\ bi^m\ s_\beta\ bi^{f(n,o)}$.

Introducînd acum comenzile $s_\beta\ bs_{\beta+1}\ b - 1$, $s_{\beta+1}\ is_{\beta+2}\ b0$, configurația devine : $bi^n\ bi^{m-1}\ s_{\beta+2}\ bbi^{f(n,o)}$. Apoi avem $bi^n\ bi^{m-1}\ s_{\beta+2}\ bb\ i^{f(n,o)}\ (T)_{\beta+2}\ bi^n\ bi^{m-1}\ s_\gamma\ bi^{f(n,o)}\ b\ (D^- * P * D^+ * P * M)_\gamma\ bi^{m-1}\ bi^{f(n,o)}\ s_\delta\ bi^n\ bi^n\ b\ (P * D^- * P * D^+)_\delta\ bi^n\ bi^{m-1}\ s_\varepsilon\ bi^{f(n,o)}\ bi^n\ b\ (MT(k))_\varepsilon\ bi^n\ bi^{m-1}\ s_\tau\ b\ i^{f(n,1)}\ b$.

Folosind comanda $s_\tau\ bs_\beta\ b0$ calculul se reia de la prima apariție a stării s_β, realizîndu-se transformarea configurațiilor de forma $bi^n\ bi^{m-j}\ s_\beta\ bi^{f(n,j)}b$ în configurații de forma $bi^n\ bi^{m-(j+1)}\ s_\tau\ bi^{f(n,j+1)}\ b$ atîta timp cît $n > j$, ajungîndu-se în cele din urmă la configurația $bi^n\ bs_\beta\ bi^{f(n,m)}\ b$, care trece în : $bi^n\ s_{\beta+1}\ b\ bi^{f(n,m)}\ b$. Folosind șirul $s_{\beta+1}\ bs_{\tau+1}\ b0$, obținem $bi^n\ s_{\tau+1}\ b\ bi^{f(n,m)}b$, care se transformă astfel :

$$bi^n\ s_{\tau+1}\ bbi^{f(n,m)}\ b\ (T)_{\tau+1}\ bi^n\ s_\lambda\ b\ i^{f(n,m)}\ bb\ (P * MT_0 * P)_\lambda\ bs_\mu\ bi^{f(n,m)}\ b\ (D^-)_\mu$$

$$s_\nu\ bbi^{f(n,m)}\ b\ (T_\nu)\ s_\omega\ bi^{f(n,m)}\ b.$$

Un ultim șir $s_\omega\ bsb0$ ne conduce la configurația $sbi^{f(n,m)}b$. $\blacksquare$

Observația II.13.4. Am demonstrat pînă acum că orice funcție primitiv-recursivă este calculabilă de o mașină Turing.

Teorema II.13.5. *Dacă funcția parțială* $f(x_1, \ldots, x_m, y)$ *este calculabilă de mașina Turing* $MT(f)$, *atunci funcția* $g(x_1, \ldots, x_m) = \mu_y(f(x_1, \ldots, x_m, y) = 0)$ *este calculabilă de o mașină Turing.*

Demonstrație. Vom da demonstrația pentru cazul $m = 1$. Pentru orice $n \in N$ avem $s_0 bi^n b (K)_0 bi^n bs_\alpha bi^n b (MT(f))_\alpha bi^n bi^0 s_\beta bi^{f(n,0)}$.

Mulțimea de șiruri $s_\beta bs_{\beta+1} b1$, $s_{\beta+1} is_{\beta+2} b0$, $s_{\beta+2} b s_{\beta+3} b - 1$, $s_{\beta+3} bs_{\beta+4} i0$, $s_{\beta+4} is_{\beta+5} i\,1$ transformă configurația $bi^n bi^j s_\beta bi^{f(x,j)}$ în configurația $bi^n bi^{j+1} s_{\beta+5} bi^{f(n,j)-1}$.

Într-adevăr, avem succesiv : $bi^n bi^j s_\beta bi^{f(n,j)} \vdash bi^n bi^j bs_{\beta+1} i^{f(n,j)} \vdash bi^n bi^j bs_{\beta+2} bi^{f(n,j)-1} \vdash bi^n bi^j s_{\beta+3} bbi^{f(n,j)-1} \vdash bi^n bi^j s_{\beta+4} ibi^{f(n,j)-1} \vdash bi^n bi^{j+1} s_{\beta+5} bi^{f(n,j)-1}$, calcul care se poate executa atîta timp cît $f(n, j) > 0$.

În continuare avem $bi^n bi^{j+1} s_{\beta+5} bi^{f(n,j)-1} (MT_0)_{\beta+5} bi^n bi^{j+1} s_\gamma b \ldots b (D^- * * D^-)_\gamma s_\delta b i^n b i^{j+1} (K)_\delta bi^n bi^{j+1} s_\varepsilon bi^n bi^{j+1} (MT(f))_\varepsilon bi^n bi^{j+1} s_\tau bi^{f(n,j+1)}$. Folosind șirul $sbsb0$, ajungem la configurația $bi^n bi^{j+1} s_\beta bi^{f(n,j+1)}$. Calculul este reluat, obținîndu-se în continuare $bi^n bi^{j+2} s_\beta bi^{f(n,j+2)}$ ș.a.m.d., atîta timp cît $f(n, j) > 0$.

Dacă la un număr dat avem $f(n, j) = 0$, obținem configurația $bi^n bi^j s_\beta bi^0 = bi^n bi^j s_\beta b$, care se transformă în $bi^n bi^j bs_{\beta+1} b$.

Folosind șirul $s_{\beta+1} bs_{\tau+1} b0$, obținem configurația $bi^n bi^j bs_{\tau+2} b$, care se transformă astfel : $bi^n bi^j bs_{\tau+1} b(D^- * D^-)_{\tau+1} bi^n s_\theta bi^j (P^* MT_0 * D^-)_\theta s_\mu bi^j$. În sfîrșit, șirul $s_\mu bsb0$ ne conduce la configurația $sbi^{g(n)}$.

Este clar că valoarea lui j pentru care s-a obținut configurația $s_\mu bi^j$ este primul număr natural, pentru care $f(n, j) = 0$ și astfel $g(n) = j$.

Dacă în procesul de calcul $f(n, j) > 0$, $\forall j \in N$ atunci mașina Turing nu se oprește niciodată, dar în acest caz nici valoarea $g(n) = \mu_y(f(n, j) = 0)$ nu-i definită. Conchidem astfel că mașina Turing construită mai sus calculează funcția g. ■

Teorema II.13.6. *Orice funcție parțial recursivă este calculabilă de o mașină Turing.*

Demonstrație. Rezultă din teoremele II.13.2 – 5. ■

Observația II.13.5. Fie E o mulțime recursivă $E \subseteq I^*$. (v. definiția I.7.11). Atunci funcția ei caracteristică χ_E este o funcție recursivă și de aceea funcția $\chi'_E : I^* \to I^*$ dată de :

$$\chi'_E(p) = \begin{cases} i, & \text{dacă } p \in E, \\ \text{nedefinită}, & \text{în caz contrar,} \end{cases}$$

unde i este un simbol fixat din I este o funcție parțial recursivă.

În baza teoremei II.13.6, această funcție parțial recursivă este calculabilă de o mașină Turing $MT = (I, S, Q; b, s_0)$. La introducerea unui cuvînt p mașina se oprește (adică trece în starea finală) atunci și numai atunci cînd

$p \in E$, ceea ce arată că limbajul E este recunoscut de mașină prin starea finală.

Prin urmare, mulțimile recursive din I^* sînt limbaje de tip 0, așa cum rezultă din teorema II.12.4.

Precizarea raportului dintre aceste două clase de limbaje va fi făcută în paragraful următor.

§ II.14. MULȚIMI RECURSIV ENUMERABILE ȘI LIMBAJE

Teorema II.14.1. *Orice limbaj de tip 0 este o mulțime recursiv enumerabilă.*

Demonstrație. Fie $L \in \mathcal{L}_0$, deci $L = L(G)$, unde $G = (I_N, I_T, x_0; F)$ este o gramatică de tip 0. Pentru orice $m \in N^*$ există numai un număr finit de deducții $D_m = \{x_0 \overset{*}{\Rightarrow} p \mid p \in I^*\}$, în care se folosesc exact m reguli de generare din F. Astfel există un algoritm în baza căruia putem alcătui lista deducțiilor $\bigcup \{D_m \mid m \in N^*\}$. Din această listă vom reține sublista conținînd deducțiile în care $p \in I_T^*$. Astfel putem alcătui o listă cu cuvintele din $L(G)$, care este deci recursiv enumerabilă (v. observația I.7.16). ■

Teorema II.14.2. *Orice limbaj de tip 1 este o mulțime recursivă.*

Demonstrație. Este suficient să arătăm că pentru un limbaj E de tip 1 există un algoritm, care ne permite să decidem dacă un cuvînt p aparține sau nu limbajului, deoarece în acest caz există un algoritm care calculează valorile funcției caracteristice χ_E. Astfel, în baza tezei lui Church, funcția χ_E va fi o funcție parțial recursivă.

Fie deci $E = L(G)$, unde G este gramatica de tip 1, $G = (I_N, I_T, x_0; F)$. Avem $e \in E$, dacă și numai dacă $(x_0 \to e) \in F$ deci, putem presupune $p \neq e$. Să considerăm șirul de cuvinte:

$$p_0 = x_0,\ p_1, \ldots, p_{n-1}, p_n = p, \text{ unde } n \geqslant 1,\ p_j \in (I_N \cup I_T)^*, \tag{II.14.1}$$

pentru $j \neq k$ avem $p_j \neq p_k$ și $l(p_j) \leqslant l(p_{j+1})$ pentru $j = \overline{0, n-1}$. Este clar că pentru un anumit p mulțimea acestor șiruri este finită și $p \in L(G)$ dacă și numai dacă există un șir (II.14.1) astfel încît:

$$x_0 = p_0 \Rightarrow p_1 \Rightarrow \ldots \Rightarrow p_{n-1} \Rightarrow p_n \in I_T^*. \tag{II.14.2}$$

Prin urmare, pentru a decide dacă $p \in L(G)$ este suficient să căutăm în mulțimea finită a șirurilor (II.14.1) un șir de forma (II.14.2).

Rămîne să observăm că pentru orice două cuvinte $p_j, p_{j+1} \in (I_N \cup I_T)^*$ se poate decide într-un număr finit de pași, dacă sau nu $p_j \Rightarrow p_{j+1}$, deoarece F este o mulțime finită. ■

Rezultatul următor arată că familia $\mathcal{L}_1$ este strict inclusă în clasa mulțimilor recursive.

Teorema II.14.3. *Există un limbaj recursiv care nu este limbaj de tip 1.*

Demonstrație. Ne vom fixa la un alfabet terminal cu doi simboli.

Fie gramatica de tip 1, $G = (I_N, \{i_1, i_2\}, x_0; F)$. Deoarece limbajul generat de gramatică nu este afectat de modul de notare a variabilelor neterminale, vom considera aceste variabile ca o parte finită a mulțimii numărabile $\{x_j \mid j \in N\}$. Să considerăm acum alfabetul $\overline{I}_T = I_T \cup \{\rightarrow, !\}$. Dacă $F = \{r_j \rightarrow p_j \mid j = \overline{1, t}\}$, mulțimea F, deci gramatica G poate fi reprezentată de cuvîntul: $r_1 \rightarrow p_1 ! p_2 \rightarrow q_2 ! \ldots ! p_n \rightarrow q_n$ din alfabetul $\overline{I}_T^*$.

Să considerăm homomorfismul $\varphi : (\overline{I}_T \cup I_N)^* \rightarrow I_T^*$ definit pe generatorii semigrupului liber $(\overline{I}_T \cup I_N)^*$ prin:

	i_1	i_2	$\rightarrow$	$!$	$x_j, j \geqslant 0$
φ	$i_2 i_1 i_2$	$i_2 i_1^2 i_2$	$i_2 i_1^3 i_2$	$i_2 i_1^4 i_2$	$i_2 i_1^{j+5} i_2$

Astfel, dacă, de pildă, $F = \{x_0 \rightarrow e,\ x_0 \rightarrow x_1 x_2,\ x_2 \rightarrow i_1,\ x_1 x_2 \rightarrow i_2 x_2\}$ atunci F este reprezentat de cuvîntul $p(G) = \varphi(x_0 \rightarrow e ! x_0 \rightarrow x_1 x_2 ! x_2 \rightarrow i_1 ! x_1 x_2 \rightarrow i_2 x_2) = i_2 i_1^5 i_2 i_2 i_1^3 i_2 i_2 i_1^4 i_2 i_2 i_1^5 i_2 i_2 i_1^3 i_2 i_2 i_1^6 i_2 i_2 i_1^7 i_2 i_2 i_1^4 i_2 i_2 i_1^7 i_2 i_2 i_1^3 i_2 i_2 i_1 i_2 i_2 i_1^4 i_2 i_2 i_1^6 i_2 i_2 i_1^7 i_2 i_2 i_1^3 i_2 i_2 i_1^2 i_2 i_2 i_1^7 i_2$.

Conchidem că oricărei gramatici de acest tip îi corespunde un cuvînt din alfabetul $\overline{I}_T^*$. Reciproc, avînd un cuvînt din $\overline{I}_T$, putem decide dacă el corespunde sau nu unei gramatici de tip 1.

Mulțimea $\overline{I}_T^*$ este numărabilă deoarece elementele ei, cuvinte de lungime finită, pot fi puse în corespondență biunivocă cu o parte a mulțimii numerelor naturale scrise în sistemul ternar atașat literelor i_1 și i_2 respectiv cifrele 1 și 2. Astfel cuvîntul $i_1 i_2 i_1 i_2 i_2$ se reprezintă ca numărul 12122 etc. Vom putea numerota deci elementele din $\overline{I}_T^*$. Procedăm inductiv, după lungimea cuvintelor. Cuvîntului vid îi atribuim numărul 1.

Să presupunem că mulțimea $\{p \mid p \in \overline{I}_T^*, l(p) \leqslant k\}$ a fost deja numerotată și că $|\{p \mid p \in \overline{I}_T^*, l(p) \leqslant k\}| = m$. Atunci celor 2^{k+1} cuvinte de lungime $k+1$ li se atribuie în ordine lexicografică[1] numerele $m+1, \ldots, m+2^{k+1}$.

Numerotarea cuvintelor din $\overline{I}_T^*$ induce o numerotare a gramaticilor de tip 1: gramatica G_j corespunde celui de-al j-lea cuvînt din $\overline{I}_T^*$, care definește o gramatică de tip 1. Această numerotare a gramaticilor este făcută cu repetiții, deoarece putem avea $G_j = G_k$ cu $j \neq k$, deoarece schimbînd ordinea regulilor de generare, obținem cuvintele $p(G_j)$, $p(G_k)$ distincte.

Fie $L = \{p_j \mid p_j \notin L(G_j)\}$. Avem $e \notin L$ deoarece $e = p_1 \in L(G_1)$ căci $G_1 = (\{x_0\}, I_T, x_0; \{x_0 \rightarrow e\})$.

[1] Cuvîntul p precede cuvîntul q în ordine lexicografică dacă $p = ri_1u$, iar $q = ri_2v$.

Afirmăm că limbajul L este recursiv, adică funcția lui caracteristică χ_L este o funcție recursivă. Conform tezei lui Church, rămîne de arătat existența unui algoritm de calcul al acestei funcții, ceea ce revine la existența unui algoritm, care pentru orice $p \in I_T^*$ este capabil să rezolve problema apartenenței lui p la L.

Fie așadar $p \in I_T^*$. Atunci există $j \in N$, astfel încît $p = p_j$. Cum gramatica G_j este de tip 1, în baza teoremei II.14.2, există un algoritm de decidere a apartenenței lui $p = p_j$ la L.

Limbajul L nu este de tip 1, deoarece dacă $L \in \mathcal{L}_1$, atunci ar exista $j \in N$, astfel ca $L = L(G_j)$. Fie al j-lea cuvînt p_j. Dacă $p \in L = L(G_j)$, atunci $p = p_j \notin L(G_j)$, ceea ce conduce la o contradicție. Dacă $p_j \notin L(G_j) = L$, atunci $p = p_j \in L$ și avem o nouă contradicție. Astfel, presupunerea $L \in \mathcal{L}_1$ este absurdă. ■

În continuare vom arăta că se poate numerota mulțimea mașinilor Turing cu alfabetul de intrare I, $|I| < \infty$, ceea ce ne va permite să arătăm în final că familia limbajelor de tip 0 pe alfabetul I este strict inclusă în $\mathcal{P}(I^*)$.

Putem presupune totdeauna că mulțimea stărilor mașinilor Turing considerate este de forma $\{s_0, s_1, ..., s_{m-1}\}$ fără a afecta calculul efectuat de mașină sau limbajul recunoscut de ea. Fie $I = \{\mathbf{b}, i_1, ..., i_{n-1}\}$.

Definiția II.14.1. *Gradul unei mașini Turing* este numărul natural $\operatorname{gr}(MT) = \max\{m, n\}$, unde $MT = (\{\mathbf{b}, i_1, ..., i_{n-1}\}, (s_0, s_1, ..., s_{m-1}), Q; \mathbf{b}, s_0)$.

Teorema II.14.4. *Mulțimea mașinilor Turing cu alfabetul de intrare I, $|I| < \infty$ este numărabilă.*

Demonstrație. Să observăm că există $3m^2n^2$, 5-uple de forma $sis'i'\alpha$, unde $s, s' \in \{s_0, ..., s_{m-1}\}$, $i, i' \in \{\mathbf{b}, i_1, ..., i_{n-1}\}$, $\alpha \in \{-1, 0, 1\}$.
Mulțimea $\mathcal{T}_k = \{MT \mid \operatorname{gr}(MT) = k\}$ este finită, deoarece cu $3m^2n^2$ 5-uple $3m^2n^2 \leqslant 3k^4$ se poate forma un număr finit de mulțimi Q distincte, și fiecare mulțime Q determină în mod unic o mașină Turing și reciproc. Astfel mulțimea $\mathcal{T}$ a mașinilor Turing este numărabilă, fiind o reuniune numărabilă de mulțimi finite.

Teorema II.14.5. *Mulțimea părților recursiv enumerabile din I^*, $\mathcal{E}(I^*)$ este numărabilă.*

Demonstrație. Într-adevăr, dacă E este o mulțime recursiv enumerabilă, există o funcție parțial recursivă $f_E : D \to N$, $D \subset N$, astfel încît $E = \{f_E(n) \mid n \in D\}$. Rezultă de aici că există o aplicație surjectivă $\varphi : \mathcal{E}(I^*) \to \mathcal{T}$, unde $\varphi(E) = MT(f_E)$. ■

Consecința II.14.1. Există părți ale lui I^* care nu sînt limbaje de tip 0, deoarece mulțimea $\mathcal{P}(I^*)$ nu-i numărabilă, în timp ce familia părților recursiv enumerabile pe același alfabet (deci, în particular și familia limbajelor de tip 0 pe I) este numărabilă.

Se știe că familia mulțimilor recursive este inclusă în familia mulțimilor recursiv enumerabile. Această incluziune este strictă, așa cum rezultă din

Teorema II.14.6. *Există o mulțime recursiv enumerabilă, care nu este recursivă.*

Demonstrație. În baza teoremei I.7.1, este suficient să construim o mulțime H recursiv enumerabilă, a cărei complementară H^c nu-i recursiv enumerabilă.

Fie $I = \{i\}$ un alfabet cu o singură literă. Dacă $E \subseteq I^*$ este un limbaj recursiv enumerabil, atunci $E = \{f(n) \mid n \in N\}$, unde f este o funcție recursivă. Există atunci un algoritm de calcul al funcției f, algoritm care se poate descrie printr-un text obișnuit, care consitituie un cuvînt (în sensul adoptat în această lucrare) asupra unui alfabet I' (de exemplu $I' = \{a,..., z, 0,...9\}$.). Cum mulțimea limbajelor recursiv enumerabile este numărabilă, mulțimea cuvintelor din I'^* care descriu algoritmii corespunzători este numărabilă. Fie ca $\{p_1, ..., p_n, ...\}$ și fie f_n funcția calculată de algoritmul descris de cuvîntul p_n.

Definim un limbaj $H \subseteq I^*$ astfel:

$$H = \{i^n \mid n \in N,\ \exists k,\ i^n = f_{n+1}(k)\}.$$

Limbajul H este recursiv enumerabil. Pentru a arăta aceasta, să notăm cu (k, j) pasul k din algoritmul descris de cuvîntul p_j. Cum orice algoritm are un număr finit de pași π_j, facem convenția că, dacă $k > \pi_j$, atunci în pasul (k, j) algoritmul nu are nici o acțiune.

Vom executa pașii diferiților algoritmi în ordinea:

$$(1, 1),\ (1, 2),\ (2, 1),\ (1, 3),\ (2, 2),\ (3, 1),\ \text{etc.}$$

În această ordine locul h ocupat de perechea (k, j) este dat de

$$h = \tau(k, j) = \frac{(k + j - 2)(k + j - 1)}{2} + k$$

Aplicația $\tau : N \times N \to N$ este bijectivă (v. exemplul I.1.11). Să considerăm acum funcția $f = D \to I^*$, $D \subseteq I^*$ dată de:

$$f(i^n) = \begin{cases} i^j, \text{ dacă } i^j \text{ este obținut în pasul } (\pi_{j+1}, j+1) \text{ și} \\ n = \tau(\pi_{j+1}, j+1) \\ \text{nedefinit, în caz contrar.} \end{cases}$$

Mulțimea valorilor funcției parțial recursive f este chiar mulțimea H, prin urmare H este o mulțime recursiv enumerabilă.

Limbajul H^c nu este recursiv enumerabil. Dacă H^c ar fi recursiv enumerabil, atunci ar exista un număr k, astfel ca H^c să fie generat de algoritmul descris de cuvîntul p_{k+1}. Pentru orice $n \geqslant 0$, cuvîntul $i^n \in H^c$ dacă și numai dacă el nu este generat de algoritmul descris de cuvîntul p_{n+1}. Prin urmare, $i^k \in H^c$ dacă și numai dacă el nu este generat de algoritmul descris de cuvîntul p_{k+1}, ceea ce contrazice presupunerea făcută. ■

§ II. 15. LIMBAJE DE TIP 0

Vom arăta că mașinile Turing au față de limbaje de tip 0 rolul de dispozitive de recunoaștere.

Teorema II.15.1. *Un limbaj de tip* 0 *este recunoscut de o mașină Turing.*

Demonstrație. Fie $G = (I_N, I_T, x_0; F)$ o gramatică de tip 0 și $L = L(G)$. Vom construi o MT nedeterministă care va accepta limbajul L. Fie $MT = (I_N \cup I_T \cup \{\mathbf{b}, \#, x\}, S, f; \mathbf{b}, s_0)$, unde $\{\mathbf{b}, \#, x\} \cap (I_N \cup I_T) = \varnothing$ cu mulțimea de stări finale $S_1 \subseteq S$, f fiind precizat ulterior.

Vom înscrie inițial pe banda mașinii un cuvînt $p \in I_T^*$. Într-o primă fază $MTND$ adaugă pe bandă înainte de cuvîntul p simbolul #, iar după cuvîntul p șirul $\# x_0 \#$. Apoi, $MTND$ fixează banda în așa fel, încît x_0 să se afle în dreptul capului de citire și se începe un proces de simulare a deducțiilor din gramatica G. Mașina are acum posibilitatea de a înscrie în locul lui x_0, între simbolii #, oricare din cuvintele $r_1, \ldots, r_n$, unde $(x_0 \to r_j) \in F$, $j = \overline{1, n}$. Apoi ea alege în mod nedeterminist o pereche de numere (h, t) cu $1 \leqslant t \leqslant \max\{l(p) \mid (p \to q) \in F\}$. Dacă cuvîntul r înscris în locul lui x_0 conține, începînd de la poziția h, un subcuvînt de lungime t care este partea stîngă a vreunei reguli de generare din F, adică $r = r' r_0 r''$ cu $l(r') = h - 1$, $l(r_0) = t$ și mulțimea $\{q_j \mid (r_0 \to q_j) \in F\} \neq \varnothing$, mașina are posibilitatea de a înlocui cuvîntul r_0 cu oricare din cuvintele q_j, deplasînd la dreapta sau la stînga cuvîntul r'' # pentru ca inserția lui q_j să se facă fără spații libere. Acest mod de a acționa se continuă pînă cînd inserarea nu mai este posibilă.

Dacă ultimul cuvînt este format numai din simboli terminali, se compară acest cuvînt, simbol cu simbol cu p, cuvînt scris inițial pe bandă. Dacă cele două cuvinte coincid, cuvîntul p este acceptat. Ca atare, un cuvînt p este recunoscut de $MTND$, dacă și numai dacă $x_0 \overset{*}{\underset{G}{\Rightarrow}} p$, adică $p \in L = L(G)$. Afirmația din enunț rezultă ținînd seama și de teorema II.12.3. ■

Teorema II.15.2. *Un limbaj este de tip* 0, *dacă și numai dacă este acceptat de o mașină Turing printr-o mulțime de stări finale.*

Demonstrație. Faptul rezultă din teorema precedentă și din teorema II.12.4. ■

Teorema II.15.3. *Familia de limbaje* $\mathcal{L}_1$ *este strict inclusă în familia de limbaje* $\mathcal{L}_0$, *adică* $\mathcal{L}_1 \subset \mathcal{L}_0$.

Demonstrație. Se știe că orice mulțime recursivă din I^* este un limbaj de tip 0 (v. observația II.13.5), iar teorema II.14.3 asigură că există un limbaj recursiv care nu este un limbaj de tip 1. ■

Am arătat în § II.2 că limbajele de tip 0 sînt închise la operațiile de reuniune, produs și închidere prin produs (v. teorema II.2.7).
Vom demonstra acum :

Teorema II.15.4. *Intersecția a două limbaje de tip* 0 *este un limbaj de tip* 0.

Demonstrație. Fie $G' = (I'_N, I'_T, x'_0; F')$ și $G'' = (I''_N, I''_T, x''_0; F'')$ două gramatici de tip 0 și $L' = L(G')$ și $L'' = L(G'')$ limbajele de tip 0, generate de aceste gramatici respectiv.

Putem presupune, fără a restrînge generalitatea că $I'_N \cap I''_N = \varnothing$.

Să considerăm gramatica:

$G = (I'_N \cup I''_N \cup I_N,\ I'_T \cup I''_T,\ X_0;\ F' \cup F'' \cup F)$, unde I_N conține pentru fiecare simbol $i' \in I'_T \cup I''_T$ un element X_i și în plus elementele X_0, X_1, X_2, X_3, iar F constă din următoarele reguli de generare:

(II.15.1) $X_0 \to X_1\, x'_0\, X_2\, X_3\, x''_0\, X_1$

(II.15.2) $X_2\, X_3\, i \to X_i\, X_3$, pentru orice $i \in I'_T \cup I''_T$

(II.15.3) $i_1\, X_i \to X_i\, i_1$, pentru toți $i, i_1 \in I'_T \cup I''_T$

(II.15.4) $X_1\, X_i\, i \to i X_1\, X_2$, pentru orice $i \in I'_T \cup I''_T$

(II.15.5) $X_2\, i \to i X_2$, pentru orice $i \in I'_T \cup I''_T$

(II.15.6) $X_1\, X_2\, X_3\, X_1 \to e$.

Prima regulă de generare care apare într-o deducție $X_0 \overset{*}{\underset{G}{\Rightarrow}} p$ este regula (II.15.1).

În continuare, aplicînd regulile de generare din F' și F'', obținem cuvîntul $r = X_1\, p'\, X_2\, X_3\, p''\, X_1$ cu $p' \in L'$ și $p'' \in L''$.

Simbolii X_1, X_2, X_3 pot fi eliminați numai prin folosirea regulii (II.15.6). În acest scop apare necesară folosirea regulilor (II.15.2 – 5).

Aplicarea acestora se poate face, cum se constată ușor, dacă și numai dacă $p' = p'' \in L' \cap L''$ și astfel obținem cuvîntul $p'\, X_1\, X_2\, X_3\, X_1$ care, folosind regula (II.15.6), trece în p', deci $L(G) = L(G') \cap L(G'')$.

Într-adevăr, fie $p' = i'_1 i'_2 \dots i'_m$, $p'' = i''_1\, i''_2 \dots i''_n$. Avem succesiv

$$X_0 \overset{*}{\Rightarrow} X_1\, i'_1\, i'_2 \dots i'_m\, X_2\, X_3\, i''_1\, i''_2 \dots i''_n\, X_1 \underset{(II.15.2)}{\Rightarrow} X_1\, i'_1 \dots i'_m\, X_{i''_1}\, X_3\, i''_2 \dots i''_n\, X_1 \overset{*}{\underset{(II.15.3)}{\Rightarrow}}$$

$$\overset{*}{\underset{(II.15.3)}{\Rightarrow}} X_1\, X_{i''_1}\, i'_1 \dots i'_m\, X_3\, i''_2 \dots i''_n\, X_1 \underset{(II.15.4)}{\Rightarrow} i'_1\, X_1\, X_2\, i'_2 \dots i'_m\, X_3\, i''_2 \dots i''_n, X_1$$

aplicarea regulii (II.14.4) putîndu-se face dacă și numai dacă $i'_1 = i''_1$.

În continuare avem: $i'_1\, X_1\, X_2\, i'_2 \dots i'_m\, X_3\, i''_2 \dots i''_n\, X_1 \underset{(II.15.5)}{\overset{*}{\Rightarrow}} i'_1\, X_1\, i'_2 \dots i'_m\, X_2\, X_3\, i''_2 \dots i''_n\, X_1$ și procedînd la fel, găsim $i''_2 = i'_2$ și obținem: $X_0 \overset{*}{\Rightarrow} i'_1 i'_2\, X_1\, i'_3 \dots i'_m\, X_2\, X_3\, i''_3 \dots i''_n\, X_1$, etc. Este evident că avem $m = n$ și $p' = p''$.

Teorema II.15.5. Fie $E \in \mathcal{L}_0$, $E \subseteq I_T$ și $u, v \notin I_T$. *Există limbajul* $E_1 \in \mathcal{L}_1$, *astfel încît*:

i) $E_1 = \{p \mid p = u^j v q,\ j > 0,\ q \in E\}$,

ii) $\forall q \in E$ *există* $j \in N^*$ *astfel ca* $u^j v q \in E_1$.

Demonstrație. Fie $E = L(G)$ cu $G = (I_N, I_T, x_0; F)$. Să considerăm gramatica $G_1 = (I_N \cup \{x_0', y\}, I_T \cup \{u, v\}, x_0', F_1 \cup F_2)$, unde $x_0', y \notin I_N$, iar $F_1 = \{r \to w \mid (r \to w) \in F, l(r) \leqslant l(w)\} \cup \{r \to y^{l(r)-l(w)} w \mid (r \to w) \in F$ și $l(r) > l(w)\}$ și $F_2 = \{x_0' \to vx_0; yv \to uv\} \cup \{ty \to yt \mid t \in I_N \cup I_T \cup \{v\}\}$.

Această gramatică este expansivă (v. definiția II.10.1) și prin urmare, limbajul $L(G_1)$ este de tip 1 (v. teorema II.10.2).

Avem $E_1 = L(G_1)$, deoarece dacă $x_0 \overset{*}{\underset{G}{\Rightarrow}} q$, atunci $x_0' \underset{G_1}{\Rightarrow} vx_0 \overset{*}{\underset{G_1}{\Rightarrow}} vy^m q \overset{*}{\underset{G_1}{\Rightarrow}} yvy^{m-1} q \underset{G_1}{\Rightarrow} yv\ y^{m-1}\ q \underset{G_1}{\Rightarrow} uyvy^{m-2} q \underset{G_1}{\Rightarrow} u^2 vy^{m-2} q \overset{*}{\underset{G_1}{\Rightarrow}} u^m v\ q$ și evident E_1 satisface condițiile *i*) și *ii*).

Teorema II.15.6. *Pentru orice limbaj de tip* 0, $E \subseteq I_T^*$ *există un limbaj de tip* 1, $E_1 \subseteq I_T^*$, *un homomorfism* $\varphi : (I_T \cup \{u, v\})^* \to I_T^*$ *și un limbaj regulat* $K \in \mathcal{R}(I_T^*)$, *astfel încît*

$$E = \varphi(E_1) = K \mid E_1, \tag{II.15.7}$$

unde am notat $K \mid E_1 = \{r \mid \exists p \in K, q \in E_1, pr = q\}$.

Demonstrație. Fie E_1 limbajul definit în demonstrarea teoremei precedente, iar φ homomorfismul definit prin:

$$\varphi(u) = \varphi(v) = e,$$

$$\varphi(i) = i, \quad \forall i \in I_T.$$

Avem evident egalitățile (II.15.7) luînd $K = \{u^m v \mid m \in N\}$. ■

Observația II.15.1. Homomorfismul φ șterge simbolii u și v din cuvintele semigrupului liber $(I_T \cup \{u, v\})$ și cum un homomorfism este o operație de substituție, se justifică observația II.10.6, substituția φ fiind singulară.

Observația II.15.2. Egalitățile (II.15.7) arată că clasa limbajelor de tip 1 nu-i închisă la homomorfisme oarecare și la operația „ $\mid$ „ cu limbaje regulate, limbajul deținut putînd fi în $\mathcal{L}_0$.

Teorema II.15.7. *Familia de limbaje* $\mathcal{L}_0$ *este închisă la operația de oglindire, adică, dacă* $E \in \mathcal{L}_0$, *atunci* $\tilde{E} \in \mathcal{L}_0$.

Demonstrație. Fie $G = (I_N, I_T, x_0; F)$ o gramatică de tip 0 și $E = L(G)$. Să considerăm gramatica $\tilde{G} = (I_N, I_T, x_0, \tilde{F})$, unde $\tilde{F} = \{\tilde{r} \to \tilde{q} \mid (r \to q) \in F\}$. Este clar că G este o gramatică de tip 0. Avem $L(\tilde{G}) = \tilde{E}$. Vom arăta că $x_0 \overset{*}{\underset{G}{\Rightarrow}} r$, dacă și numai dacă $x_0 \overset{*}{\underset{\tilde{G}}{\Rightarrow}} \tilde{r}$ procedînd prin inducție după lungimea n a deducției. Pentru $n = 1$ afirmația decurge imediat din definiția lui $\tilde{F}$. Să presupunem că ea are loc pentru n și fie deducția $x_0 \overset{*}{\underset{G}{\Rightarrow}} r_1 \underset{G}{\Rightarrow} r$ de lungime $n + 1$. Atunci, conform ipotezei inductive, avem $x_0 \overset{*}{\underset{\tilde{G}}{\Rightarrow}} \tilde{r}_1$ și în baza definiției lui $\tilde{F}$, $\tilde{r}_1 \underset{\tilde{G}}{\Rightarrow} \tilde{r}$, de unde $x_0 \overset{*}{\underset{\tilde{G}}{\Rightarrow}} \tilde{r}$ și reciproc. ■

Observația II.15.3. Argumentația de mai sus este valabilă și pentru familiile de limbaje $\mathcal{L}_1$ și $\mathcal{L}_2$. Conchidem că și aceste familii sînt închise la operația de oglindire. În ceea ce privește familia $\mathcal{L}_3$ această proprietate a fost stabilită în teorema II.4.12.

Teorema II.15.8. *Familia $\mathcal{L}_0$ este închisă la operația de substituție.*

Demonstrație. Fie $G = (I_N, I_T, x_0; F)$ o gramatică de tip 0, $I_T = \{i_1, ..., i_n\}$ și $E = L(G)$.

Să considerăm operația de substituție Φ_M generată de familia de limbaje $M = \{E_1, ..., E_n\}$, $E_j \in \mathcal{L}_0$, $E_j = L(G_1)$, unde $G_j = (I_{Nj}, I_{Tj}, x_{0j}; F_j)$, $j = \overline{1, n}$.

Fără a restrînge generalitatea, putem presupune că alfabetele neterminale sînt disjuncte două cîte două și că nici un simbol terminal nu apare în partea stîngă a vreunei reguli de generare.

Limbajul $\Phi_M(E)$ va fi generat de gramatica $G' = (I'_N, I'_T, y_0; F')$, unde $I'_N = I_N \cup \bigcup_{j=1}^{n} I_{Nj} \cup \{y_0, y, z_1, ..., z_n\}$, $I'_T = \bigcup_{j=1}^{n} I_{Tj}$, iar $F' = F^1 \cup F^2 \cup F^3 \cup \bigcup_{j=1}^{n} F_j \cup \{y_0 \to yx_0, y \to e\}$, unde F^1 este mulțimea regulilor de generare obținute din regulile de generare ale lui F, înlocuind în fiecare regulă simbolul i_j prin z_j, $j = \overline{1, n}$, $F^2 = \{yz_j \to yx_{0j} \mid j = \overline{1, n}\}$ și $F^3 = \{yi \to iy \mid i \in I_{T1} \cup ... \cup I_{Tn}\}$.

Vom arăta că $\Phi_M(E) = L(G')$. Fie $p \in \Phi_M(E)$. Există atunci $q \in E$ astfel încît p se obține din q prin înlocuirea fiecărui simbol terminal i_j printr-un cuvînt p_j din limbajul E_j. Să notăm cu q_z cuvîntul obținut din q prin înlocuirea fiecărui simbol terminal i_j prin z_j. $j = \overline{1, n}$.

În gramatica G' avem deducția:

$$y_0 \Rightarrow yx_0 \overset{*}{\Rightarrow} yq_z \Rightarrow yz_{j_1} ... z_{j_m} \Rightarrow yx_{0j_1} z_{j_2} ... z_{j_m} \overset{*}{\Rightarrow} yp_{j_1} z_{j_2} ... z_{j_m} \overset{*}{\Rightarrow} p_{j_1} yz_{j_2} ...$$

$$z_{j_m} \overset{*}{\Rightarrow} p_{j_1} yx_{0i_2} ... z_{j_m} \overset{*}{\Rightarrow} p_{j_1} y\, p_{j_2} z_{j_3} ... z_{j_m} \overset{*}{\Rightarrow} p_{j_1} p_{j_2} ... p_{j_m} y \Rightarrow p_{j_1} p_{j_2} ... p_{j_m} = p$$

și deci $p \in L(G')$.

Incluziunea inversă se arată folosind definiția mulțimii F'. ■

Imaginile prin homomorfism ale intersecțiilor de limbaje de tip 2 conduc la obținerea tuturor limbajelor de tip 0. Are loc:

Teorema II.15.9. *Orice limbaj $E \in \mathcal{L}_0$ poate fi exprimat sub forma $E = \varphi(E_1 \cap E_2)$, unde φ este un homomorfism, iar $E_1, E_2 \in \mathcal{L}_2$.*

Demonstrație. Să presupunem că $E = \mathcal{L}_0$ este generat de gramatica de tip 0, $G = (I_N, I_T, x_0; F)$, unde $F = \{r_i \to q_j \mid j = \overline{1, k}\}$. Să notăm $I = I_N \cup I_T$ și fie alfabetele $I_{T'} = \{i' \mid i \in I_T\}$, $I_{yz} = \{y, z_1, ..., z_k\}$.

Să notăm cu p' cuvîntul obținut din $p \in I_T^*$ prin înlocuirea fiecărui simbol i prin i'. Prin convenție $e' = e$. Fie alfabetul:

$I_1 = I_N \cup I_T \cup I'_T \cup I_{yz}$ și $\varphi: I_1^* \to I_T^*$ homomorfismul definit prin:

$$\varphi(u) = \begin{cases} i, & \text{dacă } u = i' \in I_T \\ e, & \text{dacă } u \in I_N \cup I_T \cup I_{yz}. \end{cases}$$

Să considerăm limbajele :

$$E_1 = (\{r_1\, p_j\, z_j\, r_2\, y \widetilde{r_2}\, \widetilde{q_j}\, \widetilde{r_1}\, y \mid r_1, r_2 \in I^*,\ j = \overline{1,k}\}^* \setminus \{e\})\,(I_T')^*$$

$E_2 = \{x_0\, z_j\, y \mid j = \overline{1,k}\}\,\{\widetilde{r_1\, r_2\, y\, r_1\, z_h\, r_2\, y} \mid r_1, r_2 \in I_T^*,\ h = \overline{1,k}\}^* \{pyp' ,\ p \in I_T^*\}$. Aceste limbaje sînt de tip 2. Într-adevăr E_1 este de forma $E_1 = (H^* \setminus \{e\})\,.\,(I_T')^*$ și va fi deci suficient să verificăm că H este un limbaj de tip 2. Limbajul H este generat de gramatica de tip 2, $G'' = (\{v_0, v_1, v_2\}, I_1, v_0$; $(v_0 \to v_1\, y,\ v_1 \to xv_1\, x,\ v_1 \to p_j\, z_j\, v_2\, \widetilde{q_j},\ v_2 \to xv_2\, x,\ v_2 \to y \mid x \in I,\ j = \overline{1,k}\}$, așa cum se constată imediat. La fel se arată că limbajul E_2 este de tip 2.

Să arătăm că $E = \varphi(E_1 \cap E_2)$. Fie $p \in E$. Deoarece $E = L(G)$ există un $m \in N$, cuvintele $r_{j_1}, r_{j_2} \in I^*$ și indicii $g(j)$ cu $g(j) = \overline{1,k}$ definit pentru $j \in \overline{0,m}$ astfel încît :

$r_{j1}\, q_{g(j)}\, r_{j2} = r_{(j+1)1}\, p_{g(j+1)}\, r_{(j+1)2}$, unde $r_{01} = r_{02} = e$, $p_{g(0)} = x_0$ și $r_{m1}\, q_{g(m)}\, r_{m2} = p$, adică cuvîntul p este generat prin deducția : $x_0 = r_{01}\, p_{g(0)}\, r_{02} \Rightarrow r_{01}\, q_{g(0)}\, r_{02} = r_{11}\, q_{g(1)}\, r_{12} \Rightarrow r_{11}\, q_{g(1)}\, r_{12} = r_{21}\, p_{g(2)}\, r_{22} \overset{*}{\Rightarrow} r_{(m-1)1}\, q_{g(m-1)2}\, r_{(m-1)2} = r_{m1}\, p_{g(m)}\, r_{m2} \Rightarrow r_{m1}\, q_{g(m)}\, r_{m2} = p$.

Să considerăm aplicația $\theta : I^* \times \{1,\ldots,k\} \times I^* \to I_1^*$ dată prin $\theta(r_1, j, r_2) = r_1\, p_j\, z_j\, r_2\, y\, \widetilde{r_2}\, \widetilde{q_j}\, \widetilde{r_1}\, y$ și fie cuvîntul $r = \theta(r_{01}, g(0), r_{02})\,\theta(r_{11}, g(1), r_{12}) \ldots \theta(r_{m1}, g(m), r_{m2})\,p'$, avem $r \in E_1$.

Fie aplicația $\sigma : I^* \times \{1,\ldots,k\} \times I^* \to I_1^*$ dată de $\sigma(r_1, j, r_2) = \widetilde{r_1\, r_2\, y\, r_1\, z_j\, r_2\, y}$. Putem scrie $r = x_0\, z_{g(0)}\, y\ r_{01}\, q_{g(0)}\, r_{02}\, y\ r_{11}\, p_{g(1)}\, z_{g(1)}\, r_{12}\, y \ldots r_{(m-1)1}\, q_{g(m-1)}\, r_{(m-1)2}\, y r_{m1}\, p_{g(m)}\, z_{g(m)}\, r_{m2}\, y.\ r_{m1}\, q_{g(m)}\, r_{m2}\, y = x_0\, z_{g(0)}\, y\, r_{11}\, p_{g(1)}\, r_{12}\, y\, r_{11}\, p_{g(1)}\, z_{g(1)}\, r_{12}\, y \ldots r_{m1}\, p_{g(m)}\, r_{m2}\, y\, r_{m1}\, p_{g(m)}\, z_{g(m)}\, r_{m2}\, y\, r_{m1}\, q_{g(m)}\, r_{m2}\, yp' = x_0\, z_{g(0)}\, y\, \sigma(r_{11}\, p_{g(1)}, g(1), r_{12}) \ldots \sigma(r_{m1}\, p_{g(m)}, g(m)\, r_{m2})\, p\, yp'$, de unde $r \in E_2$, deci $r \in E_1 \cap E_2$. Cum $\varphi(r) = p$ rezultă că $p \in (E_1 \cap E_2)$.

Invers, fie $p \in \varphi(E_1 \cap E_2)$. Există atunci $r \in E_1 \cap E_2$ astfel ca $p = \varphi(r)$. Deoarece $r \in E_1$ există m, $g(j) \in N$ și $r_{jh} \in I_1^*$ cu

(II.15.8) $\quad r = \theta(r_{01}, g(0), r_{02})\,\theta(r_{11}, g(1), r_{12}) \ldots \theta(r_{m1}, g(m), r_{m2})\,p'$.

Dar, cum $r \in E_2$ există n, $f(j) \in N$ și $r_{jh} \in I_1^*$ cu

(II.15.9.) $\quad r = x_0\, z_{f(0)}\, y\, \sigma(r_{11}, f(1), r_{12}) \ldots \sigma(r_{n1}, f(n), r_{n2})\, \widetilde{pyp'}$.

Se observă că $m = n$ prin calcularea numărului de apariții ale lui y în membrii drepți ai egalităților (II.15.8.–9.). Apoi avem $g(j) = f(j)$, $j = \overline{0,m}$ și $r_{01} = r_{02} = e$, $p_{g(0)} = x_0$, $r_{j1}\, p_{g(j)} = r_{j1}$, $r_{j2} = r_{j2}$, $r_{(j-1)1}\, q_{g(j-1)}\, r_{(j-1)2} = r_{j1}\, r_{j2} = r_{j1}\, p_{g(j)}\, r_{j2}\, j = \overline{1,m}$ și $p = r_{m1}\, q_{g(m)}\, r_{m2}$.

Obținem astfel următoarea deducție în gramatica G :

$$x_0 = r_{01}\, p_{g(0)}\, r_{02} \Rightarrow r_{01}\, q_{g(0)}\, r_{02} = r_{11}\, p_{g(1)}\, r_{12} \overset{*}{\Rightarrow} r_{(m-1)1}\, q_{g(m-1)}\, r_{(m-1)2} = r_{m1}\, p_{g(m)}\, r_{m2} \Rightarrow r_{m1}\, q_{g(m)}\, r_{m2} = p,$$

prin urmare $p \in E$. ■

B I B L I O G R A F I E [1)]

ÁBRAHÁM, S.

[1] Some questions of phrase structure grammars. Computational linguistics **4** (1965), 61—70.

ABRAMSON, H.D.

[1] A note on left — recursive rules and partitioning of a recognition matrix for syntax-directed translation. Nordisk. Tidskr. Informations behandling (BIT) (1970), 1—5.

AHO, A.

[1] Indexed grammars — An extension of context — free grammars. J. Assoc. Comput. Mach. **15** (1968), 617—671.

ALTMAN, E., B.; BANERJI, R., B.

[1] Some problems of finite representability. Information Control **8** (1965), 251—263.

AMAR, V.; PUTZOLU, G.

[1] Generalization of regular events. Information Control **8** (1965), 56—63.

АНИСИМОВ, А. В.

[1] О групповых языках. Кибернетика, Киев, 1971, № 4, стр. 18—24.

[2] Веспереборный анализ языков програмирования методы развертки. Proceedings of Symopsium and Summer School of Mathematical Foundations of Computer Science, High Tatras 1973, p. 175 — 180.

ARBIB, M., A.

[1] The algebraic theory of machines. languages and semigroups. Academic Press, New-York — London, 1968.

BALTUSH, F.

[1] Maşini matematice şi gramatici generative. Stud. Cerc. Mat. **21**. (1969), 831—876.

BAR-HILLEL Y.

[1] Language and information. Addison—Wesley Publ. Comp., Reading, Mass., 1964.

BÄR, D.

[1] Zur Theorie der syntaktischen Analysealgorithmen für kontextfeie Grammatiken I. Elektron. Informationsverarbeit **Kybernetik 5** (1969), 171—205.

[2] Zur Theorie der syntaktischen Analysealgorithmen für kontextfreie Grammatiken II. Elektron. Informationsverarbeit Kybernetik **5** (1969), 217—238.

BARBAULT, M. C.; DESCLÉS, J. P.

[1] Vers une formalisation des grammaires transformationnelles. Math. Sci. Humaines No 34 (1971), 27—41, No 35 (1971).

BARTHOLOMES, F.; HOTZ, G.

[1] Homomorphismen und Reduktionen linearer Sprachen. Springer — Verlag, Berlin — New York, 1970.

БЕЛЕЦКИЙ, М. И.

[1] Бесконтекстные и доминационные грамматики и связанные с ними алгорифмические проблемы. Кибернетика Киев, 1967, № 4, стр. 90—97.

[1)] Se va consulta şi bibliografia din : CREANGĂ, I.; REISCHER, C.; SIMOVICI, D.: Introducere algebrică în informatică. * Teoria automatelor. Editura Junimea, Iaşi, 1973.

[2] Математическая модель предложения. Кибернетика Киев, 1971, № 2, стр. 125—132.
[3] Об одном способе описания управления в категориальных грамматиках. Кибернетика Киев 1971, № 4, стр. 146—147.
BELL, J. L.; SLOMSON, A. B.
[1] Models and Ultraproducts. North Holland Publ. Amsterdam, 1969.
BIRKHOFF, G.
[1] Lattice theory. Amer. Math. Soc. Colloquium Publications Providence, Rhode Island, 1968.
BLUM, M.
[1] A machine — independent theory of recursive functions. J. ACM **14** (1967), 322—336.
BOASSON, L.
[1] Cônes rationnels et familles agréables de langages. — Application au langage à compteur. Thesis, Univ. de Paris, 1971.
BOASSON, L. NIVAC, M.
[1] Transductions et familles de langages. Math. Sci. Humaines No 35, (1971), 31—37.
БОДНАРЧУК, В. Г.
[1] Автоматы и события. Укр. Матем. Журнал **14** (1962) 351—361.
[2] Системы уравнений в алгебре событий. Ж. вычисл. матем. и матем. физика **3** (1963) 1077—1088.
[3] Метрическое пространстве событий. Кибернетика (Киев) 1965, № 1, 24—27, № 4, 22—30.
BOOK, R.
[1] Time-bounded grammars and their language.
J. Comput. System. Sci. **5** (1971) 397—429.
[2] Terminal context in context-sensitive grammars. Center for Res. in Comput. Technol. Harvard Univ. Preprint no. 10 (1971).
BOOK, R.; GINSBURG, S.
[1] Multi-steck-counter languages. Math. Systems Theory **6** (1972), 37—48.
BOOK, R. V.; GREIBACH, S. A.
Quasi-realtime languages. Math. Systems Theory **4** (1970), 97—111.
BOOK, R. V.; GREIBACH, S. A.; WAGBREIT, B.
[1] Time — and tape — bounded Turing acceptors and AFLs.
J. Comput. System. Sci. **4** (1970). 606—621.
BOOK, R.; EVEN, S.; GREIBACH, S.; OTT, G.
[1] Ambiguity in graphs and expressions.
IEEE Trans. Comput. C — **20** (1971) 144—153.
BOOK, R., V.; GREIBACH, S., A.; IBARRA, O. H.; WEGBREIT, B.
[1] Tape-bounded Turing acceptors and principal AFLs., J. Comput. System. Sci. **4** (1970), 622—625.
BOURBAKI, N.
[1] Théorie des ensembles · Éléments de mathématique. Hermann, Paris, 1970.
BOUCHER, C.
[1] Fermeture de certaines classes de langages formels sous des parmutations lingvistiques. Rev. Française Informat. Recherche Operationnelle **3** (1969), Sér. R. — 2, 51—61.
BRAINERD, B.
[1] An analog of a theorem about context free languages.
Information Control **11** (1968), 561—567.
[2] Introduction to the mathematics of language study. American Elsevier Publ. Co., New York, 1971.
BURKHARD, W. A.; VARAIYA, P. P.
[1] Complexity problems in real time languages.
Information Sci. **3** (1971), 87—100.
CANTOR, D., C.
[1] On the ambiguity problem of Backus systems. J. ACM **9** (1962), 477—479.
CARACCIOLO, A.
[1] Some remarks on the syntax of symbolic programming languages.
Com. ACM **6** (1963), 456—460.

CĂRĂUȘU, A.
[1] On a natural ordering of rules in context-sensitive grammars. Preprint of the XIth International Congress of Lingvists. Bologna — Firenze, 1972, 1229—1236.
[2] On ordered rewriting systems. Cahiers de ling. théor. et appl. **9** (1972), 277—284.
CARLUCCI, L.
[1] A formal system for texture languages. Pattern Recognation **4** (1972), no. 1, 53—72.
CHOMSKY, N.
[1] A note on phrase structure grammars. Information Control **2** (1959), 393—395.
CÎȘLARU, C.
[1] Configurații de ordin infinit. Stud. Cerc. Mat. **21** (1969), 561—570.
COCHET, Y.; NIVAT, M.
[1] Une généralisation des ensembles de Dyck. Israel J. Math. **9** (1971), 389—395.
COLMERAUER, A.
[1] Total precedence relations. J. Assoc. Comput. Mach. **17** (1970), 14—30.
CONWAY, J. M.
[1] Regular algebra and finite machines. Chapman and Hall, London, 1971.
CRĂCIUN, C. V.
[1] Δ-domination and conditioned domination in algebraic grammar. Stud. Cerc. Mat. **20** (1968), 683—695.
[2] Unele proprietăți ale operatorilor contextuali din lingvistica algebrică. Stud. Cerc. Mat. **22** (1970), 419—434.
КРЮКОВ, Ю. А.
[1] Машины Тюринга с тремя состояниями и двумя симболами и одним состоянием и н симболами. Кибернетика Киев 1971, № 1, стр. 12—13.
CREANGĂ, I.; REISCHER, C.; SIMOVICI, D.
[1] Introducere algebrică în informatică * Teoria automatelor. Editura Junimea, Iași, 1973.
CUDIA, D.; SINGLETARY, W.
[1] Degrees of unsolvability in formal grammars. J. Assoc. Comput. Mach. **15** (1968) 680, 692.
ÇULIK, K. II.
[1] n-ary grammars and the description of mapping of langauges. Kybernetika (Prague) **6** (1970), 99—117.
DAVIS, M.
[1] Computability and unsolvability. Mc Graw-Hill, New York, 1963.
DINCĂ, AL.
[1] Ecuații diofantice și limbaje context-free. Studii Cerc. Mat. **22** (1970). 1059—1172.
[2] Distanțe contextuale în lingvistica algebrică. Studii Cerc. Mat. **25** (1973), 223—265.
[3] Distanțe și diametre într-un semigrup (cu aplicații la teoria limbajelor). Studii Cerc. Mat. **25** (1973), 359—378.
DUYVESTEYN, A. J. W.
[1] Sentence analysis. Math. Centrum. Amsterdam Afd. Zuivere Wisk. 1969, ZW—005.
EICKEL, J.; LOECKX, J.
[1] The relation between derivations and syntactical structures in phrase-structure grammars. J. Comput. System Sci. **6** (1972), 267—282.
ELGOT, C., C.
[1] Review of "A remark on finite transducers", IRE Trans. Electron. comput. EC **11** (1962), 802.
ENESCU, GH.
[1] Logică simbolică, Ed. științifică, Buc. 1971.
ERSHOV, A., P.
[1] Programming Programme for the BESM Computer. Pergamon Press, New York, 1959.
FABIAN, V.
[1] Structural unambiguity of formal languages. Czech. Math. J. **14** (1964), 394—430.

FISCHER, M.
[1] Two characterizations of context sensitive languages. IEEE Conf. Record 10 th Ann. Symp. Switching Automata Theory (1969), 149–156.
FLIESS, M.
[1] Transductions algebriques. Rev. Française Informat. Recherche Operationelle **4** (1970) Sér. R – 1, 109–125.
FLOYD, R. W.
[1] On the nonexistence of a phrase structure grammar for ALGOL **60**. CACM. **5** (1962), 483–484.
[2] On ambiguity in phrase structure languages. CACM **5** (1962), 526–534.
[3] Syntactic analysis and operator precedence. J. ACM **10**, (1963), 316–333.
[4] Bounded context syntactic analysis. CACM **7**, (1964), 62–67.
[5] New proofs and old theorems in logic and formal linguistics. Computer Associates Inc., Wakefield, Mass, 1964.
[6] The syntax of programming languages – a survey. IGEC, **13** (1964) 346–353.
FRIANT, J.
[1] Grammaires ordonnées – grammaires matricielles. Univ. de Montréal MA – 101 (1968).
[2] Langages ultralinéaires et superlinéaires. Nouvelles caractérisations. Univ. de Montréal MA – 102 (1968).
FRIS, I.
[1] On stop – conditions in the definitions of constructive languages. Zeitscher. f. Math. Logik und Grundlagen d. Math. **11** (1965), 61–73.
[2] Grammars with partial ordering of the rules. Information Control **12** (1968), 415–425.
FUJII, M.; KASANI, T.
[1] Some structural equivalence relations and transformation of context-free grammars. Electron Commun. Japan **52** (1969), 162–170.
FUJITA, T.
[1] A note on recursive language. Information Precessing in Japan **8** (1968), 12–13.
GABRELIAN, A.
[1] Pure grammars and pure languages. Univ. of Waterloo, Dep. of Appl. Anal. and Comput. Sci. Res. Rep. CSRR 2027 (1970).
GAIFMAN, H.
[1] Dependency systems and phrase structure systems. Information Control **8** (1965), 304–337
GILL, A.
[1] Introduction to the theory of finite-state machines. Mc Graw Hill, New York, 1962.
GINSBURG, S.
[1] An introduction to mathematical machine theory. Addison–Wesley, Reading, Mass. 1962.
[2] The mathematical theory of context-free languages. Mc Graw-Hill Book Co., New York, 1966.
GINSBURG, S., GREIBACH, S.
[1] Abstract families of languages. Studies in Abstract Families of Languages. Mem. Amer. Math. Soc. Nº 87, 1–32. Amer. Math. Soc. Providence, R. I. 1969.
[2] Principal AFL. J. Comput. System. Sci. **4** (1970), 308–338.
GINSBURG, S.; HARRISON, M. A.
[1] Bracketed context free languages. System Devel. Comp. Rept. TM–738/023/00. Jan. 4, 1966.
[2] Bracketed context-free languages. J. Comput. System Sci. **1** (1967), 1–23.
GINSBURG, S.; HOPCROFT, J.
[1] Images of AFL under certain families of homomorphisms. Math. Systems Theory **5** (1971), 216–227.
GINSBURG, S.; PARTEE, B.
[1] A mathematical model of transformationel grammars. Information Control **15** (1969), 297–334.
GINSBURG, S.; ROSE, G.F.
[1] On the existence of generators for certain AFL. Information Sci. **2** (1970), 431–446.
GINSBURG, S.; SPANIER, E.H.
[1] Mappings of languages by two-tape devices. J. ACM, **12** (1965), 423–434.
[2] Control sets on grammars. Math. Systems Th. **2** (1968), 159–177.
[3] Derivation-bounded languages. J. Comput. Sci. **2** (1968), 228–250.

[4] Substitution in families of languages. Information Sci. **2** (1970), 83—110.
[5] AFL with the semilinear property. J. Comput. System. Sci. **5** (1971), 365—396.
GINSBURG, S.; GREIBACH, S.; HOPCROFT, J.
[1] Pre — AFL. Studies in Abstract Families of Languages. Mem. Amer. Math. Soc.. № 87, 41—51. Amer. Math. Soc. Providence, R.I., 1969.
GINSBURG, S.; HIBBARD, T. N.; ULLIAN, J. S.
[1] Sequences in context free languages. III. J. Math., **9** (1965), 321—337.
GINZBURG, A.
[1] Algebraic theory of automata. Academic Press, New York, 1968.
ГЛАДКИЙ, А. В.
[1] Формальные грамматики и языки, Москва, 1973.
ГЛАДКИЙ, А. В.; МЕЛЬЧУК, И. А.
[1] Элементы математической лингвистики. Наука, Москва, 1969.
ГЛЕБСКИЙ, Ю. В., КОГАН, Д. И.
[1] Аддитивно управляемые системы и языки: некоторые алгорифимические проблемы. Кибернетика Киев 1971, № 4, стр. 25—38.
GREIBACH, S. A.
[1] Formal parsing systems. C. AC M **7** (1964), 499—504.
[2] A note on code sets and context free languages. Math. Lingvistics and Automatic Translation, Harvard Univ. Computation Lab. Rept. NSF 15, 1965, II—1—II—5.
[3] A new normal form theorem for context-free phrase structure grammars. **J. ACM, 12** (1965), 42—52.
[4] A note on undecidable properties of formal languages. Math. Systems Theory **2** (1968), 1—6.
[5] An infinite hierarchy of context-free languages. J. Assoc. Comput. Mach. **16** (1969), 91—106.
[6] Full AFL's and nested iterated substitution. IEEE conf. Record 10th Ann. Symp. Switching Automata Theory (1969), 222—230.
[7] Characteristic and ultrarealtime languages. Information Control **18** (1971), 65—98.
GREIBACH, S.; GINSBURG, S.
[1] Multitape AFA. J. Assoc. Comput. Mach. **19** (1972), 193—221.
GREIBACH, S., HOPCROFT, J.
[1] Scattered context grammars. J. Comput. System Sci. **3** (1969), 233—247.
[2] Independence of AFL operations. Studies in Abstract Families of Languages. Mem. Amer. Math. Soc. No. 87. 35—40. Amer. Math. Soc. Providence, R.I., 1969.
GRIFFITHS, T. V.; PETRICK, S. R.
[1] On the relative efficiencies of contex-free grammar recognizers. C. ACM, **8** (1965), 289—300.
GROSS, M.; LENTIN, A.
[1] Introduction to formal grammars. A Springer-Verlag, New York-Berlin, 1970.
GRUSKA, J.
[1] On structural unambiguity of formal languages. Czech. Math. J. **15** (1965), 283—293.
[2] Introduction in formal languages: Some properties of reducing transformations and of isolable sets. Czech. Math. J. **15** (1965), 406—414.
[3] A characterization of context free languages. J. Comput. System Sci. **5** (1971), 353—364.
[4] A few remarks on the index of context-free grammars and languages. Information Control **19** (1971), 216—223.
GRZEGORCZYK, A.
[1] Some classes of recursive functions. Rozprawy Mat. **4** (1953), 1—45.
HARRISON, M., A.; SCHKOLNICK, M.
[1] A grammatical characterization of one-way nondeterministic stack languages. **J. Assoc. Comput. Mach. 18** (1971), 148—172.
HAVEL, I.
[1] On one — side context — sensitive grammars. In J. Dörr and G. Hotz (eds). Automaten theorie und formale Sprachen. Bibliographisches Inst. Mannheim, 1970, 221—225.
[2] Strict deterministic languages. Univ. of California (Berkeley), Comput. Sci. Dept. Tech. Rep. No. 1, 1971.
HERMAN, G. T.
[1] Linear regular languages. I. Acta Cybernet. **1** (1969), 3—12.

[2] The computing ability of a developmental model for filamentous organisms. J. Theoret. Biol. **25** (1969), 421–424.
[3] Closure properties of some families of languages associated with biological systems. State Univ. of New York at Buffalo, Dept. of Comput. Sci. Rep. 7–71. 1971.
HOTZ, G.; SCHNORR, C.P. (eds)
[1] Tagung über Formale Sprachen. Gesellschaft für Mathematik und Datenverarbeitung Bonn, 1970.
IBARRA, O. H.
[1] Simple matrix languages. Information Control **17** (1970), 359–391.
IRONS, E.T.
[1] A syntax directed compiler for ALGOL 60 CACM, **4** (1961), 51–55.
JÄRVI, T.
[1] On control sets induced by grammars. Ann. Acad. Sci. Eenn. Ser. A, I, (1970), 7.
JONES, N. D.
[1] Notes on the theory of Computability. Pennsylvania State Unis, 1968.
[2] Context-free languages and rudimentary attributes. Math. System Theory **3** (1969), 102–109.
[3] A note on the index of a context-free language. Information Control **16** (1970), 201–202.
JURCA, M.
[1] Asupra relației dintre tipul și Q-clasa unui cuvînt. Studii. Cerc. Mat. **22** (1970), 1033–1040.
KAMEDA, T.; VOLLMAR, R.
[1] Note on tape reversal complexity of languages. Information Control **17** (1970), 203–215.
KAMINGER, F.
[1] The noncomputability of the channel capacity of context-sensitive languages. Information Control **17** (1970), 175–182.
KASAI, T.
[1] An hierarchy between context free and context-sensitive languages. J. Comput. System Sci. **4** (1970), 492–508.
KASAMI, T.; TANIGUCHI, K.
[1] Some decision problems of formal languages. Electron. Commun. Japan. **51** (1968), 151–158.
KLEENE, S.C.
[1] Introduction to metamathematics. D. Van Nostrand, Princeton, New Jersey, 1952.
KNUTH, D.E.
[1] Top-down syntax analysis. Acta Informatica **1** (1970), 79–110.
KOBAYASHI, K.
[1] Some unsolvable problems concerning context-free languages and their application to on-line language recognizers. Electron. Commun. Japan **51** (1968), 93–97.
KOPRIVA, J.
[1] Decomposition translations and syntax translation schemata. Kybernetika (Prague) **6** (1970), 403–417.
KOSTER, C. H. A.
[1] Two-level grammars. Seminar on Automata Theory and Mathematical Lingvistics. Utrecht, Abstract no. 11 (1970).
KRÁL, J.
[1] On multiple grammars. Kybernetika **1** (1969), 60–85.
[2] A modification of a substitution theorem and some necessary and sufficient conditions for sets to be context-free. Math. System. Theory **4** (1970), 129–139.
KUICH, W.
[1] Systems of pushdown acceptors and context-free grammars. Elektron. Informationsverarbeit. Kybernetik **6** (1970), 95–113.
[2] A context-free language and enumeration problems on infinite trees and digraphs. J. Combinatorial Theory Ser. B. **10** (1971), 135–142.
KUICH, W.; MAURER, H.
[1] On the inherent ambiguity of simple tuple languages. Computing (Arch. Elektron. Rechnen) **7** (1971), 194–203.
KURKI-SUONIO, R.
[1] Notes on top-down languages. BIT **9** (1969), 225–238.

[2] A programmer's introduction to computability and formal languages. Studentlitteratur, Sweden, 1971.
LEARNER, A. ; LIM, A.L.
[1] A note on transforming context-free grammars to Wirth-Weber precedence form. Comput. J. **13** (1970), 142—144.
LEHMANN, D.
[1] *L.R* (*k*) grammars and deterministic languages. Israel, J. Math. **10** (1971), 526—530.
ЛЕТИЧЕВСКИЙ, А. А.
[1] Представление контекстне-свободных языков в автоматах с памятью типа pushdown. Кибернетика, Киев **1**, (1965) 80—84.
[2] О последовательном расширении алгоритмических языков. Кибернетика, Киев 1967, № 6, стр. 1—6.
LEWIS, P. M. II ; STEARNS, R.
[1] Syntax directed transduction. J. Assoc. Comput. Mach. **15** (1968), 464—488.
LINDENMAYER, A.
[1] Mathematical models for cellular interactions in development, parts I—II. J. Theoret. Biol. **18**, (1968), 280—315.
LOECKX, J.
[1] Mechanical construction of bounded-context parsers for Chomsky-o-type languages. Philips Res. Rep. Suppl. Nr. 3 (1964) V + 92 p.
[2] Grammaires formalles et automates. Bull. Soc. Math. Belg. **21** (1969), 145—165.
[3] The parsing for general phrase — structure grammars. Information Control **16** (1970), 443—461.
LYNCH, W. C.
[1] Ambiguities in Backus form languages. Doctoral dissertation, Univ. of Wisconsin, Madison Wisc. 1963.
MAGIDOR. M. . MORAN, G.
[1] Probabilistic tree automata and context free languages. J. Math. **8** (1970), 340—348.
MAL'CEV, A.J.
[1] Algebraiceskie sistemy. Nauka, Moskva, 1970.
МАЛЬЦЕВ, А. И.
[1] Алгоритмы и рекурсивные финкции. Изд. «Наука» физ—мат. литературы. Москва, 1965.
MALUSZYNSKI, T.
[1] The lattice of subgrammars of context free grammars.
Algorytmy **7** (1970), 21—27.
MARCU, E.T.
[1] On an operator that is related to Sestier's operator in contextual algebraic analysis. Stud. Cerc. Mat. **21** (1969), 499—508.
MARCUS, S.
[1] Gramatici și automate finite. Edit. Academiei R.S.R., București, 1964,
[2] Contextual grammars. Rev. Rumaine Math. Pures Appl. **14** (1969), 1525—1534.
MATTHEWS, G. H.
[1] Discontinuity and asymmetry in phrase structure grammars.
Information Control **6** (1963), 137—146.
[2] A note on asymmetry in phrase structure grammars. Information Control, **7** (1964), 360—365.
[3] Two-way languages. Information Control **10** (1967), 111—119.
MAURER, H.
[1] Theoretische Grundlagen der Programmiersprachen. Hochschul taschenbücher 404, Bibliographisches Inst., 1969.
[2] A direct proof of the inherent ambiguity of a simple context-free languages. J. Assoc. Comput. Mach., **16** (1969), 256—260.
MAYER, O.
[1] Some restrictive devices for context-tree grammars. Information Control **20** (1972). 69—92.
Mc NAUGHTON, R.; PAPERT, S.
[1] Counter-free automata. Research Monograph no. 65. M.T.I. Press, Cambridge, Mas. 1971.

McWHIRTER, I.
[1] Substitution expressions, J. Comput. System Sci. **5** (1971), 629–637.
MEALY, G. H.
[1] A method for synthetizing sequential circuits. Bell. Syst. Tech. J. **34** (1955) 1045–1079.
MIHALYI, M.
[1] Despre noţiunea de productivitate sintactică. Stud. Cerc. Mat. **21** (1969), 419–431.
MILGRAM, D. L.; ROSENFELD, A.
[1] A note on scattered context grammars. Information Precessing Lett. **1** (1971/1972), 47–50.
MINSKY, M.; PAPERT, S.
[1] Unrecognizable sets of numbers. J. Assoc. Comput. Mach. **13** (1966), 281–286.
MIZUMOTO, M.; TOYODA, J.; TANAKA, K.
[1] Some considerations on fuzzy automata. Electron. Commun. Japan **52** (1969), 93–99.
MORSE, M.: HEDLUNG, G.
[1] Symbolic dynamics. Amer. J. Math. **60** (1938), 815–866.
NASH, B.; COHEN, R.
[1] Parallel leveled grammars. IEEE Conf. Record 10 th Ann. Symp. Switching Automata Theory (1969), 263–276.
NASU, M.; HONDA, N.
[1] A context-free language which is not acceptable by a probabilistic automaton. Information Control **18** (1971), 233–236.
NAVRATIL, E.
[1] Context-free grammars with regular conditions. Kybernetika (Prague) **6** (1970), 118–126.
NEBERSKY, L.
[1] On one of the boundaries of the Constituent conception of grammar, Kybernetika (Prague) **7** (1971), 227–229.
NISHIZAWA, T.
[1] An algebraic proof of Kleene-Myhill's equivalence theorem. Mem. Fac. Sci. Kyushu Univ. Ser. A, **24** (1970), 94–99.
[2] A note on an ordering of nonrepeated word sequences. Mem. Fac. Sci. Kyushu Univ. Ser. A, **24** (1970), 101–109.
[3] Sequential machines with general counters. Mem. Fac. Sci. Kyushu Univ. Ser. A, **24** (1970), 110–119.
NIVAT, M.
[1] Transduction des langages de Chomsky. Ann. Inst. Fourier, Grenoble **18** (1968), 339–455.
NOVOTNY, M.
[1] Algebraic structures of mathematical lingvistics. Bull. Math. Soc. Sci. Math. R. S. Roumaine (N.S.) **12** {**60**} (1968), 87–101.
[2] On configurations in the sense of Kulagina.
Spisy Prirod. Fak. Univ. Brno, 1969, 301–315.
[3] On two classes of languages. Spisy Prirod. Fak. Univ. Brno, 1970. Cislo 518, 465–470 (1971).
[4] Complete characterization of classes of Chomsky by means of configurations. Acta F.R.N. Univ. Comen. – Mathematica – Mimoriadne, cislo (1971), 63–71.
[5] On some relations defined by languages. Prague Studies in Math. Linguistics **4** (1970), 157–170.
[6] On homomorphism of norm-bounded languages. Prague Studies in Math. Linguistics **5** (1973).
[7] On some operators reducing generalized grammars. Matematicky ustav CSAV, pobočka Brno, 1973.
OGDEN, W.
[1] A helpful result for proving inherent ambiguity. Math. Systeme Theory **2** (1968), 191–194.
ORMAN, G.
[1] O condiţie de existenţă a mulţimilor omologe. Bul. Inst. Politehn. Braşov, Ser. A. Mec. **11** (1969), 24–27.
[2] Considérations sur les langages „context-free" linéairs à gauche et a droite. Rev. Roumaine Math. Pures Appl. **16** (1971), 1107–1111.
PAIR, C.
[1] Sur les notions algébriques liées à l'analyse syntaxique. Rev. Française Informat. Operationnelle **4** (1970) Ser. R-3, 3–29.

PAULL, M.; UNGER, S.
[1] Structural equivalence of context free grammars. J. Comput. System Sci, **2** (1968), 427—163.
PAZ, A.
[1] Formal series, finiteness properties and decision problems. Ann. Acad. Sci. Fenn. Ser. A, I No 493 (1971), 16 p.
PETER, R.
[1] Bemerkungen zu den G. Révérzschen terminalen Grammatiken. Ann. Univ. Sci. Budapest. Eötrös Sect. Math. **11** (1968), 109—113.
PETERS, P. S. Jr.; RITCHIE, R.
[1] On restricting the base component of transformational grammars. Information Control **18** (1971), 483—501.
PETRONE, L.
[1] On precedence grammars. Olivetti Electron. Res. Lab. Rept. Febr. 1965.
POPESCU, E.
[1] Asupra unui model algebric al părții de vorbire. Studii și Cerc. Mat. **23** (1971), 1151—1160.
POST, E. L.
[1] Formal reduction of the general combinatorial decision problem. Amer. J. Math. **65** (1943), 197—215.
RAISCHI, C. T.
[1] Relația de dominare și analiza algebrică contextuală. Stud. Cerc. Mat. **20** (1968), 31—41.
[2] Contexte caracteristice în lingvistica algebrică.
Studii și cercetări Mat. **21** (1969), 123—139.
[3] Operatorii lui Nebesky și contextele caracteristice.
Stud. Cerc. Mat. **22** (1970), 891—916.
[1] Absolutely parallel grammars and two-way finite-state transducers. J. Comput. System Sci. **6** (1972), 321—312.
РЕДЬКО, В. Н.
[1] К синтаксическому анализу контекстне свободных языков. Кибернетика Киев 1966, № 3, стр. 52—63.
REISCHER, C.; SIMOVICI, D.
[1] Limbaje litere de context. Universitatea „Al. I. Cuza" — Iași, Centrul de multiplicare, 1973.
RÉVÉSZ, GY.
[1] Syntactic analysis and unilateral context-sensitive grammars. Studia Sci. Math. Hungar, **4** (1969), 267—278.
[2] Unilateral context sensitive grammars and left-to-right parsing. J. Comput. System. Sci. **5** (1971), 337—352.
[3] Dual pushdown automata and context sensitive grammars. Acta Cybernet. **1** (1972), 147—157.
ROGERS, H.
[1] Theory of recursive functions and effective computability. McGraw Hill Book Co. New York, 1967.
ROSENBERG, A. L.
[1] A machine realization of the linear context-free languages. Information Control **10** (1967), 175—188.
[2] A note on ambiguity of context-free languages and presentations of semilinear sets. J.A.C.M. **17** (1970), 44—50.
ROSENFELD, A.
[1] Isotonic grammars, parallel grammars, and picture grammars, Machine Intelligence, **6**, 281—294, American Elsevier, New York, 1971.
ROSENKRANTZ, D.
[1] Programmed grammars and classes formal languages. J. Assoc. Comput. Mach. **16**, (1969), 107—131.
ROSENKRANTZ, D.J.; STEARNS, R.E.
[1] Properties of deterministic top-down grammars.
Information Control **17** (1970), 226—256.

ROZENBERG, G.
[1] On the introduction of orderings into the grammars of Chomsky's hierarchy. Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys. **17** (1969), 559—563.
[2] T O L — systems and languages. ERCU Publicaties (Utrecht) no, 103, 1971.
[3] On OL-systems with restricted use of productions. ERCU Publicaties (Utrecht) no. 116, 1971.
[4] L-systems with interactions ERCU Publicaties (Utrecht) no. 120, 1971.
[5] Direction controlled programmed grammars. Acta Informatica **1** (1972), 242—252.

ROZENBERG, G.; DOUCET, P.
[1] On OL-languages. Information Control **19** (1971), 302—318.

SALOMAA, A.
[1] On finite automata with a time-variant structure. Information Control **13** (1968), 85—98.
[2] On grammars with restricted, use of productions. Ann. Acad. Sci. Fennicae, Ser. AI **454** (1969).
[3] On the index of a context-free grammar and language. Information Control **14** (1969), 474—477.
[4] Periodically time-variant context free grammars. Information Control **17** (1970), 294—311.
[5] Matrix grammars with a liftmost restriction. Information Control **20** (1972), 143—149.
[6] Formal languages. Academic Press. New York, London, 1973.

SALWICKI, A.
[1] Formalised algorithmic languages. Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys. **18** (1970), 227—232.
[2] On the equivalence of FS-expressions and programs. Bull. Acad. Polon. Sci. Sér. Sci. Math. Astronom. Phys. **18** (1970), 275—278.

САМОЙЛЕНКО, Л. Г.
[1] О свойствах грамматик непосредственных составляющих. Кибернетика Киев, 1966, № 5, стр. 24—27.
[2] О методе построения и свойствах контекстнозависимых грамматик и языков. Proceedings of Symposium and Summer School of Mathematical Foundations of Computer Science, High Tatras 1973, p. 307 — 312.

SCHORRE, D. V.
[1] A necessary and sufficient condition for a context-free grammar to be unambiguous. System Development Corp. Rept. SP-2153, 1965,

SCHÜTZEMBERGER, M. P.
[1] Some remarks in Chomsky's context-free languages. M. I. T. Res. Lab. Electron. Quart. Prag. Rept., 1961.
[2] On a theorem of R. Jungen. Proc. Amer. Math. Soc. **13** (1962), 885—890.
[3] Classification of Chomsky's languages. Proc. IFIP. Working Conference on Formal Language Description Languages. Baden, Austria, 1964.

SHAMIR, E.
[1] A remark on discovery algorithms for grammars. Information Control **5** (1962), 246—251.
[2] Mathematical Models of Languages. Proceedings of IFIP Congress 65, Spartan Borks, Inc. Washington, D. C. 1965, 71—75.
[3] Some inherently ambiguous context-free languages. Information Control **18** (1971), 355—363.

SIMOVICI, D. A.
[1] On some measures on free semigroups induced by semiautomate. Proceedings of Symposium and Summer of Mathematical Foundations of Computer Science, High Tatras 1973, p. 313—316.

SIROMONEY R.: SIROMONEY, G.
[1] Equal matrix languages and Kleene closure. Proc. Indian Acad. Sci. Sect. A **74** (1971), 172—188.

SKOWRON, A.
[1] A syntactic translation of the machine-determined languages. Bull. Acad. Polon. Sci. Math. Astronom. Phys. **18** (1970), 603—607.
[2] Semantic translation of programming languages. Z. Math. Logik Grundlagen Math. **17** (1971), 39—46.
[3] Languages determined by machines systems. Bull. Acad. Polon. Sci. Ser. Sci. Math. Astronom. Phys. **19** (1971), 321—326.
SMITH, W. B.
[1] Error detection in formal languages. J. Comput. System Sci. **4** (1970), 385—405.
SMULLYAN, R.M.
[1] Theory of formal systems. Princeton Univ. Press. Princeton, New Jersey, 1961.
SOLOMON, S. Y.
[1] Lingvistica algebrică și teoria modelelor. Stud. Cerc. Mat. **21** (1969), 1007—1134.
SZÖCS, H.
[1] O metodă de transformare a proceselor structural omogene în mulțimi statistice. Stud. Cerc. Mat. **20** (1968), 259—265.
ШРЕЙДЕР, Ю. А.
[1] О вариационных принципах в лингвистике. Кибернетика Киев, 1966, № 2, стр. 49—56.
TANAKA, E.; KASAI, T.; TAKI, I.
[1] Self-learning of finite state language by machines. Electron. Commun. Japan **52** (1969), 115—122.
TANIGUCHI, K.; KASAMI, T.
[1] Reduction of context-free grammars. Information Control **17** (1970), 92—108.
THATCHER, J. W.
[1] Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. J. Comput., System Sci. I (1967), 317—322.
THIERRIN, G.
[1] Decomposition des langages réguliers. Rev. Française Informat. Recherche Operationnelle **3** (1969), Ser. R—3, 45—50.
THUE, A.
[1] Probleme über Veränderungen von Zeichenreihen nach gegebenen Regeln. Skrifter utgit av Videnskapsselskapet i Kristiana I **10** (1914), 34 p.
ULLIAN, J.S.
[1] Failure of a conjecture about context free languages. Information Control **9** (1966), 61—65.
ULLMAN, J. D.
[1] Pushdown automata with bounded backtrack. System Devol. Corp. Rept. TM — 738/022/00 Dec. 1965.
UREONEN, T.
[1] On axiom systems for regular expressions and on equations involving languages. Ann. Univ. Turku. Ser. A I No 145 (1971), 52 pp.
VAIDA, D.
[1] Algoritmi de compilare — cu aplicații la limbaje de tip ALGOL. Editura didactică și pedagogică, București, 1971.
VERE, S.
[1] Translation equations. Comm. ACM **13** (1970), 83—89.
WALTER, H
[1] Verallgemeinerte Pullbackkonstruktion bei Semi-Thuesystemen und Grammatiken. Elektron. Informationsverarbeit. Kybernetic **6** (1970), 239—254.
WALTERS, D.
[1] Deterministic context-sensitive languages. I and II. Information Control **17** (1970), 14—61.
WEGBREIT, B.
[1] A generator of context-sensitive languages. J. Comput. System. Sci. **3** (1969), 456—461.
WOOD, D.
[1] A note on top-down deterministic languages. Nordisk. Tidskr. Informationstebehaudling (BIT) **9** (1969), 387—399.
[2] The normal form theorem — another proof. Comput. **12** (1969), 139—147.

[3] A generalized normal form theorem for context-free grammars. Comput. J. **13** (1970), 272–277.
YNTEMA, M.
[1] Inclusion relations among families of context-free languages. Information Control **10** (1967), 572–597.
[2] Cap expressions for context-free languages. Information Control **18**, (1971), 311–318.
ZADED, L. A.
[1] Quantitative fuzzy semantics. Information Sci. **3** (1971), 159–176.
ZELINKA, B.
[1] Sur le PS-isomorphisme fort des langages. Bull. Math. Soc. Sci. Math. R.S. Roumanie (N.S.) **13** (61) (1969), 123–126.
[2] Les graphes de Hasse des modèles analitiques de langages. Bull. Math. Soc. Sci. Math. R.S. Roumanie (N.S.) **13 61** (1969), 389–392.
[3] Le groupe des S-automorphismes d'un langage. Rev. Roumaine Math. Pures Appl. **15** (1970), 453–458.
[4] Deux problèmes sur les grammaires à un nombre d'etats fini. Rev. Roumaine Math. Pures Appl. **16** (1971), 999–1002.
ZIMMER, R.
[1] Soft precedence. Information Processing Lett. **1** (1971), 108–110.

INDICE ALFABETIC

Alfabet 28
alfabet de ieşire 73
alfabet de intrare 73
alfabet neterminal 65
alfabet terminal 65
algebră Boole 26
algebră Boole B_2 26
algebră Lindenbaum 58
aplicaţie 11
aplicaţie bijectivă 11
aplicaţie contactantă 32
aplicaţie de marcare 18
aplicaţie injectivă 11
aplicaţie surjectivă 11
arbore 17
arbore de deducţie 86
arc în graf 16
arc în hipergraf 18
aritatea unei operaţii 11
automat determinist 73
automat de tip Leticevski 123
automat finit 73
automat iniţial 73
automat liniar mărginit 141
automat nedeterminist 73
automat neiniţial 73
automat pushdown 113
automat slab iniţial 73
axiomele calculului propoziţional 55

Bijecţie 11
buclă 17

Calcul al unei configuraţii 151
cap de citire-scriere 113, 150
cel mai mare element, 13
cel mai mic element 13
clasă de echivalenţă 13
coeficient al unui term 17
componentă conexă 17
componentă conexă a unui hipergraf 19
complementara unei mulţimi 11
compunerea de maşini Turing 165
concatenare 28
conectiv logic 54
configuraţie instantanee a unui *ALM* 142
configuraţie instantanee a unui *APD* 114
configuraţie instantanee a unei *MT* 151
configuraţie instantanee a unei *MTBI* 154
configuraţie instantanee a unei *MTMMB* 158
configuraţie instantanee precede o configuraţie instantanee într-un *ALM* 142
configuraţie instantanee precede o configuraţie instantanee într-un *APD* 114
configuraţie instantanee precede o configuraţie instantanee într-o *MT* 151

configurație instantanee terminală 164
congruență 28
congruență la dreapta 28
congruență la stînga 28
corespondența biunivocă 11
corpul *GF* (2) 26
cuantificator existențial 58
cuvînt 28
cuvînt generat direct într-o gramatică 65
cuvînt generat într-o gramatică 66
cuvîntul nul 28
cuvînt recunoscut de un automat prin un semnal de ieșire 75

Deducție 66
deducție imediată la stînga 90
deducție imediată la dreapta 90
derivare la stînga în raport cu *i* 124
descendent al unui vîrf 17
descendent direct al unui vîrf 17
descendent imediat 17
descendent impropriu 17
descendent în hiperarbore 19
descendent propriu 17
diferența de mulțimi 11
diferența simetrică de mulțimi 11
dispozitiv de comandă 113, 150
drum de lungime *n* într-un hipergraf 19
drum într-un graf 16

Echivalența nucleară a unei aplicații 16
ecuație universală 37
element complementar 25
extremitate finală a unui arc 16
extremitate inițială a unui arc 16

Forma normală a lui N. Chomsky 92
forma normală a lui S. Greibach 97
formulă deschisă 59
formulă inferabilă 60
formulă valabilă 62
formulă universal valabilă 62
funcție calculabilă de o mașină Turing 165
funcție caracteristică 14
funcție de ieșire 73
funcție de proiecție 44, 51
funcție de succesiune 44, 51
funcție de trecere 73
funcție F-parțial recursivă 49
funcție nulă 44, 51
funcție numerică parțială 44
funcție parțială 44
funcție parțial calculabilă de o mașină Turing 165
funcție parțial recursivă
funcție primară 61
funcție primară pe alfabetul I
funcție primitiv-recursivă 45
funcție standard 108
funcție totală calculabilă de o mașină Turing 165
funcție total definită 44
funcție vectorială standard 109

Gradul unei mașini Turing 172
graf conex 17
graf marcat 18
graf orientat 16
graf parțial 16
graf tare conex 17
gramatică 65
gramatică „context free" 68
gramatică „context sensitive" 68
gramatică cu marcaj final # 133
gramatică de ordin *n* 134
gramatică de tip O 67
gramatică de tip 1 67
gramatică de tip 1 cu marcaj final # 133
gramatică de tip 2 67
gramatică de tip 2 ambiguă 91
gramatică de tip 2 neambiguă 91
gramatică de tip 2 cu proprietatea de autoimersiune 103
gramatică de tip 3 68

gramatică expansivă 131
gramatică liberă de context 68
gramatică liniar-mărginită 135
gramatică sensibilă la context 68
gramatici echivalente 66

Hiperarbore 19
hiperarbore de deducție 137
hipergraf 18

Infimul unei mulțimi parțial ordonate 13
injecție 11
intersecție de mulțimi 11
înălțimea unui vîrf 17
închiderea prin produs a unui limbaj 29
închiderea reflexivă și tranzitivă a unei relații 16

Juxtapunere 28

Latice 22
latice complementată 26
latice completă 22
latice distributivă 26
latice mărginită 25
limbaj acceptat de un ALM prin o mulțime de stări finale 142
limbaj acceptat de o MT prin o mulțime de stări finale 152
limbaj acceptat de o $MTBI$ prin o mulțime de stări finale 155
limbaj acceptat de o $MTMMB$ prin o mulțime de stări finale 158
limbaj acceptat de o $MTND$ prin o mulțime de stări finale 161
limbaj context free" 68
limbaj „context 68
sensitive"
limbaj descris de un term 108
limbaj de tip ALGOL 110
limbaj de tip Dyck 56
limbaj de tip 2 cu proprietatea de autoimersiune 103
limbaj generat de o gramatică 66
limbaj inerent ambiguu 91
limbaj neambiguu 91
limbaj n-parantetic 66
limbaj pe alfabetul I 29
limbaj recunoscut de un APD 115
limbaj regulat 76
limbaj reprezentat într-un automat inițial prin o mulțime de stări 75
limbaj reprezentat într-un automat inițial prin un semnal de ieșire 75
limbaj reprezentat într-un APD prin o mulțime de stări finale 116
limbaj reprezentat într-un AL 123
limbaj reprezentat într-un semiautomat nedeterminist 76
limbajul formulelor calculului predicatelor 58
limbajul formulelor calculului propozițional 54
literă 28
lungimea unui calcul 114
lungimea unui cuvînt 28
lungimea unui drum 17
lungimea unei demonstrații 56

Mașină Turing 150
mașină Turing bilateral infinită 154
mașină Turing cu n benzi 157
mașină Turing nedeterministă 159
mașină Turing nestaționară 151
memorie de tip pushdown 113
metalimbaj 107
metrică 29
metrică indusă pe o normă 30
modus ponens 58
monoid 27
monoid semilaticial 28
monom ciclic 39

monom k-ciclic 39
monom strict ciclic 39
mulțimea arcelor 16
mulțimea funcțiilor F-elementare 44
mulțimea funcțiilor F-recursive 45
mulțimea simbolilor de marcare 18
mulțimea vidă 10
mulțime 10
mulțime cît 13
mulțime de stări 73
mulțime finită 20
mulțime infinit numărabilă 20
mulțime numărabilă 20
mulțime parțial ordonată 12
mulțime parțial recursivă 49
mulțime primitiv recursivă 49
mulțime recursivă 52
mulțime recursiv enumerabilă 52
mulțimi egale 10

Normă 30
norma unui vector 30
număr cardinal 20

Operator atașat unei stări 75
operator de i-minimizare 52
operator de i-recursie 51
operator de închidere 15
operator de minimizare 48
operator de recursie primitivă 44
operator de superpoziție 44
operator indus de un limbaj 34
operator nesingular 34
operație binară 12
operație de închidere prin prefixe 81
operație de oglindire 81
operație de substituție generată de o mulțime de limbaje 82
operație de substituție nesingulară generată de o mulțime de limbaje 138
operație n-ară 11
operator contractant 32

Parte 10
partiție 13
polinom liniar 38
polinom pe $\mathcal{P}(I^*)$ 38
principiul dualității
pentru latici 22
produs cartezian de mulțimi 11
produs de limbaje 29
produs de relații 12
programul unei mașini Turing 164
proiecția j-a produsului cartezian 11

Rădăcina unui arbore 17
rădăcina unui hiperarbore 19
rangul unei relații 13
realizare a calculului propozițiilor 55
regulă de generare 65
relație antisimetrică 12
relație binară 11
relație de echivalență 12
relație de ordine 12
relație de preordine 12
relație diagonală 12
relație inversă 12
relație n-ară 11
relație reflexivă 12
relație simetrică 12
relație tranzitivă 12
relațiile lui De Morgan 26
reprezentant al unei clase de echivalență 13
reuniune de mulțimi 11

Semiautomat atașat unui automat 73
semigrup 27
semigrup cu unitate 27
semigrup liber 28
semigrup semilaticial 28, 39
semilatice 27
semn de punctuație 54
simbol 28
simbol de bază al calculului propozițiilor 54

simbol de start 114
simbol metalingvistic 107
sistem de închidere 14
situaţie de oprire pentru o MT 151
soluţie minimală 37
sp - algebră 124
sp - algebră generată de o familie de limbaje 124
sp - algebră secvenţială 124
spaţiu liniar normat 30
spaţiu liniar real 29
spaţiu metric 29
spaţiu metric complet 31
stare globală a unui automat pushdown 114
stare terminală a unei mulţimi subarbore de deducţie 87
subgraf 16
subgraf parţial 16
submulţime 10
supremumul unei mulţimi parţial ordonată 13
surjecţie 11
Şir Cauchy 31
şir convergent 31

Tautologie 55
teorema de punct fix a lui Banach 31
teorema de punct fix a lui Knäster-Tarski 25
teorema de completitudine a calculului propoziţional 58
term 108
teza lui A. Church 49

Variabilă a unui term 108
vîrf al unui graf 16
vîrf al unui hipergraf 18
vîrf conectat al unui hipergraf 19
vîrf interior al unui hipergraf 19
vîrf terminal al unui hiperarbore 19

TABLA DE MATERII

Prefață 5

Listă de notații principale 6

Partea I Elemente de algebră

§ I.0 Introducere 10
§ I.1. Noțiuni de teoria mulțimilor 10
§ I.2. Latici 22
§ I.3. Semigrupuri 27
§ I.4. Spații metrice și spații liniare 29
§ I.5. Spațiul metric al limbajelor 32
§ I.6. Ecuații pe mulțimea limbajelor 38
§ I.7. Funcții recursive 44
§ I.8. Elemente de logică matematică 54

Partea a II-a Limbaje formale

§ II.0. Introducere 65
§ II.1. Gramatici 65
§ II.2. Ierarhia gramaticilor 67
§ II.3. Automate finite și limbaje regulate 72
§ II.4. Limbaje de tip 3 76
§ II.5. Limbaje de tip 2 (limbaje litere de context) 85
§ II.6. Caracterizarea limbajelor de tip 3 în clasa limbajelor de tip 2. 102
§ II.7. Limbaje de tip ALGOL. 106
§ II.8. Automate pushdown (automate cu pile de memorie) 112
§ II.9. Automate de tip Leticevski 122
§ II.10. Limbaje de tip 1 (limbaje sensibile la context) 131
§ II.11. Automate liniar mărginite 141
§ II.12. Mașini Turing — dispozitive de recunoaștere a limbajelor 150
§ II.13. Mașini Turing — dispozitive de calcul 164
§ II.14. Mulțimi recursiv enumerabile și limbaje. 170
§ II.15. Limbaje de tip 0. 174

Bibliografie 179

Indice alfabetic 191

CONTENTS

— Preface 5

— List of main notations 6

I. **Elements of Algebra**

§ I.0. Introduction 10
§ I.1. Notions of set theory 10
§ I.2. Lattices 22
§ I.3. Semigroups 27
§ I.4. Metric spaces and linear spaces 29
§ I.5. The metric space of the languages 32
§ I.6. Equations on the set of languages 38
§ I.7. Recursive functions 44
§ I.8. Notions of mathematical logic 54

II. **Formal languages**

§ II.0. Introduction 65
§ II.1. Grammars 65
§ II.2. The hierarchy of grammars 67
§ II.3. Finite automata and regular languages 72
§ II.4. Type — 3 languages 76
§ II.5. Type — 2 languages (context-free languages) 85
§ II.6. Characterization of type — 3 languages in the class of type — 2 languages. 102
§ II.7. ALGOL — like languages 106
§ II.8. Pushdown automata 112
§ II.9. Leticevsky automata 122
§ II.10. Type — 1 languages (context-sensitive languages) 131
§ II.11. Linear-bounded automata 141
§ II.12. Turing machines as recognition devices for languages. 150
§ II.13. Turing machines as computation devices 164
§ II.14. Recursively enumerable sets and languages 170
§ II.15. Type — 0 languages 174

References 179

Alphabetic index 191

TABLE DES MATIÈRES

Préface 5
Liste des symboles usuels 6

Première partie. Elements d'algèbre

§ I.0. Introduction 10
§ I.1. Notions de théorie d'ensembles 10
§ I.2. Treillis 22
§ I.3. Semigroupes 27
§ I.4. Espaces metriques et espaces linéaires 29
§ I.5. L'espace métrique des langages 32
§ I.6. Equations sur l'ensemble des langages 38
§ I.7. Fonctions récursives 44
§ I.8. Notions de logique mathématique 54

Deuxième partie (langages formales)

§ II.0. Introduction 65
§ II.1. Grammaires 65
§ II.2. Hierarchie des grammaires 67
§ II.3. Automates finis et langages réguliers. 72
§ II.4. Langages de type — 3 76
§ II.5. Langages de type — 2 (langages libres de contexte) 85
§ II.6. Caractérisations des langages de type 3 dans la classe des langages de type 2 102
§ II.7. ALGOL — like langages 106
§ II.8. Automates pushdown (Automates à pile de memoire) 112
§ II.9. Automates de type Leticevsky 122
§ II.10. Langages de type — 1 (langages sensibles au contexte) 131
§ II.11. Automates linéairement bornés 141
§ II.12. Machines de Turing-Dispositifs pour reconnaitre les langages 150
§ II.13. Machines de Turing. Dispositifs de calcul. 164
§ II.14. Ensembles recursifs dénombrables et langages 170
II.15. Langages de type — 0. 174

Bibliographie 179

Index alphabétique 191

ОГЛАВЛЕНИЕ

— Предисловие 5

— Список главных обозначений 6

Часть первая — Агебраические повятия

§ I.0. Введение 10
§ I.1. Понятия теории множеств 10
§ I.2. Структуры 22
§ I.3. Полугруппы 27
§ I.4. Метрические и линиарные пространства 29
§ I.5. Метрическое пространство языков 32
§ I.6. Уровнения на множества языков 38
§ I.7. Рекурсивные функции 44
§ I.8. Понятия математической логики 54

Часть вторая — Формальные языки

§ II.5. Введение 65
§ II.1. Грамматики 65
§ II.2. Иерархия языков 67
§ II.3. Конечные автоматы и регулярные языки 72
§ II.4. Языки типа 3 76
§ II.5. Языки типа 2 85
§ II.6. О характеризование языков типа 3 в классах языков типа 2 . . 102
§ II.7. Языки типа АЛГОЛ 106
§ II.8. Pushdown автоматы 112
§ II.9. Автоматы типа Летичевский 122
§ II.10. Языки типа 1 (языки чувственные на контексте) 131
§ II.11. Линиар-ограниченные автоматы 141
§ II.12. Машины Тьюринга — устройства узнавания языков 150
§ II.13. Машины Тьюринга — устройства исчилении 164
§ II.14. Рекурсивно-перечислимые множества и языки 170
§ II.15. Языки типа 0. 174

Библиография 179

Алфавитный указатель 191

Redactor : MIHAIL GRĂDINARU
Tehnoredactor : MIHAI BUJDEI

Apărut 1974. Format 70×100/16. Coli tipo 12,50. Bun de tipar la 21-VIII-1974. Tiraj : 6 200 ex. broşate.
EDITURA „JUNIMEA" — IAŞI, ROMÂNIA

Tiparul executat la Întreprinderea poligrafică Iaşi, str. Vasile Alecsandri nr. 6, sub cd. nr. 174
Republica Socialistă România

EDITURA JUNIMEA
IAȘI

Lei 11,50